(19) BUNDESREPUBLIK DEUTSCHLAND

DEUTSCHES PATENT- UND MARKENAMT

(12) **Offenlegungsschrift**

(10) **DE 100 14 448 A 1**

(51) Int. Cl.[7]:
G 06 F 15/173
G 06 F 13/12

(21) Aktenzeichen: 100 14 448.9
(22) Anmeldetag: 23. 3. 2000
(43) Offenlegungstag: 16. 11. 2000

2

DE 100 14 448 A 1

(30) Unionspriorität:

09/276,428	25. 03. 1999	US
09/346,592	02. 07. 1999	US
09/347,042	02. 07. 1999	US
09/455,106	06. 12. 1999	US
09/482,213	12. 01. 2000	US

(71) Anmelder:
Dell USA, L.P., Round Rock, Tex., US

(74) Vertreter:
Patent- und Rechtsanwälte Bardehle, Pagenberg, Dost, Altenburg, Geissler, Isenbruck, 81679 München

(72) Erfinder:
Altmaier, Joseph, Riverside, Ia., US; Harris jun., George W., Mountain View, Calif., US; Lane, Jerry Parker, San Jose, Calif., US; Legueux jun., Richard A., Hudson, N.H., US; Merrell, Alan R., Fremont, Calif., US; Nespor, Jeffrey S., Pleasanton, Calif., US; Nolan, Shari J., San Jose, Calif., US; Panas, Michael G., Hayward, Calif., US; Parks, Ronald L., Danville, Calif., US; Taylor, James A., Livermore, Calif., US; Taylor, Alastair, San Jose, Calif., US

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

Prüfungsantrag gem. § 44 PatG ist gestellt

(54) Speicherverwaltungssystem

(57) Ein Speicherbereichs-Verwaltungssystem unterstützt Speicherbereiche. Der Speicherserver umfaßt eine Vielzahl von Kommunikationsschnittstellen. Eine erste Gruppe von Kommunikationsschnittstellen aus der Vielzahl ist geeignet zur Verbindung mit allen Arten von Anwendern von Daten. Eine zweite Gruppe von Kommunikationsschnittstellen aus der Vielzahl ist geeignet für die Verbindung zu entsprechenden Geräten in einem Pool von Speichergeräten zur Verwendung in einem Speicherbereich. Datenverarbeitungsressourcen in dem Server werden mit der Vielzahl von Kommunikationsschnittstellen verbunden, um Daten unter den Schnittstellen zu transferieren. Die Datenverarbeitungsressourcen umfassen eine Vielzahl von Treibermodulen und konfigurierbarer Logik, die Treibermodule zu Datenpfaden verbindet. Jeder konfigurierte Datenpfad dient als eine virtuelle Verbindung, der eine Gruppe von Treibermodulen, die aus der Vielzahl von Treibermodulen ausgewählt worden sind, umfaßt. Ein Datenspeichervorgang, der an einer Kommunikationsschnittstelle empfangen wird, wird auf einen der konfigurierten Datenpfade abgebildet. Eine Anzeige und ein Anwendereingabegerät sind in den datenverarbeitenden Strukturen enthalten, um Bilder, die auf der Anzeige angezeigt werden, zu verwalten.

DE 100 14 448 A 1

Beschreibung

HINTERGRUND DER ERFINDUNG

Bereich der Erfindung

Die vorliegende Erfindung betrifft den Bereich von Massenspeichersystemen. Insbesondere betrifft sie die Verwaltung von Speichervorgängen in intelligenten Speicherbereichsnetzwerken und deren Konfiguration.

Der Stand der Technik

Das Speichern von großen Datenmengen in sogenannten Massenspeichersystemen ist im Begriff, eine übliche Vorgehensweise zu werden. Massenspeichersysteme umfassen typischerweise Speichergeräte, die mit Dateiservern oder Datennetzwerken verbunden sind. Anwender in dem Netzwerk kommunizieren mit den Datenservern, um Zugriff auf die Daten zu erhalten. Die Dateiserver sind typischerweise über Datenkanäle mit spezifischen Speichergeräten verbunden. Die Datenkanäle werden üblicherweise mit Point-to-Point-Kommunikationsprotokollen implementiert, die ausgelegt sind für die Verwaltung von Speichervorgängen.

In dem Maße, wie die Menge an Speicher zunimmt und die Anzahl der Dateiserver im Kommunikationsnetzwerk wächst, ist das Konzept eines Speicherbereichsnetzwerkes (storage area network, SAN) aufgekommen. Speicherbereichsnetzwerke verbinden eine Anzahl von Massenspeichersystemen in einem Kommunikationsnetzwerk, das für Speichervorgänge optimiert ist. Beispielsweise werden durch einen Glasfaserkanal vermittelte Schleifennetzwerke (fibre channel arbitrated loop networks, FC-AL) als SANs implementiert. Die SANs unterstützen viele Point-to-Point-Kommunikationssitzungen zwischen den Anwendern des Speichersystems und den spezifischen Speichersystemen in dem SAN.

Dateiserver und andere Anwender der Speichersysteme werden konfiguriert, um mit spezifischen Speichermedien zu kommunizieren. Wenn die Speichersysteme expandieren oder ein Medium in dem System ausgetauscht wird, ist eine Rekonfiguration bei den Dateiservern und anderen Anwendern notwendig. Wenn die Notwendigkeit auftritt, Daten von einem Gerät auf ein anderes in einem sogenannten Datenmigrationsvorgang zu verschieben, ist es ferner häufig notwendig, den Zugriff auf die Daten während des Migrationsvorganges zu blockieren. Nachdem die Migration abgeschlossen ist, muß die Rekonfiguration beim Anwendersystem ausgeführt werden, damit die Daten von dem neuen Gerät verfügbar werden.

Insgesamt vermehren sich in dem Maße wie die Komplexität und die Größe der Speichersysteme und Netzwerke zunimmt, auch die Probleme der Verwaltung der Konfiguration der Anwender der Daten und der Speichersysteme selbst. Demgemäß sind Systeme notwendig, die die Verwaltung der Speichersysteme vereinfachen und gleichzeitig die Vorteile der Flexibilität und Möglichkeiten der SAN-Architektur benutzen.

ZUSAMMENFASSUNG DER ERFINDUNG

Die vorliegende Erfindung schafft Systeme und Verfahren zur Speicherbereichsverwaltung. Speicherbereichsverwaltung ist eine zentralisierte und sichere Verwaltungsfähigkeit, die auf existierenden Hardwareinfrastrukturen eines Speicherbereichsnetzwerkes angeordnet ist, um eine hohe Leistungsfähigkeit, eine hohe Verfügbarkeit und fortgeschrittene Speicherverwaltungsfunktionalität für heterogene Umgebungen zu schaffen. Speicherbereichsverwaltung schafft einen Kern eines robusten Gefüges eines Speicherbereichsnetzwerkes, das übernommene und neue Ausrüstung integrieren kann, Netzwerk- und Speicherverwaltungsaufgaben den Servern und Speicherressourcen abnehmen kann und Netzwerkbasierte Anwendungen aufnehmen kann, so daß sie über alle Komponenten des Speicherbereichsnetzwerkes verteilt werden können. Speicherbereichsverwaltung ermöglicht das Erzeugen und Optimieren einer heterogenen Umgebung eines Speicherbereichsnetzwerkes, die bei der Verwendung von Systemen und Techniken aus dem Stand der Technik nicht zur Verfügung steht.

Die vorliegende Erfindung schafft ein System für die Verwaltung von Speicherressourcen in einem Speichernetzwerk nach Speicherbereichen (Domains). Das System umfaßt eine Vielzahl von Kommunikationsschnittstellen beziehungsweise Interfaces, die für eine Verbindung mit Clients, Speichersystemen und dem Speichernetzwerk über Kommunikationsmedien geeignet sind. Eine Verarbeitungseinheit ist mit der Vielzahl von Kommunikationsschnittstellen verbunden und umfaßt Logik, um eine Gruppe von Speicherorten aus einem oder mehreren Speichersystemen in dem Netzwerk als einen Speicherbereich für eine Gruppe von zumindest einem Client aus dem einen oder mehreren Clients in dem Speichernetzwerk zu konfigurieren. Das System umfaßt in verschiedenen Kombinationen Elemente zum Bereitstellen von Multiprotokollunterstützung über die Vielzahl der Kommunikationsschnittstellen hinweg, Logik zum Routen von Speichervorgängen innerhalb eines Speicherbereiches in Antwort auf Vorgangsidentifizierer, die innerhalb der Protokolle enthalten sind, eine Verwaltungsschnittstelle zum Konfigurieren der Speicherbereiche, Logik zum Übersetzen eines Speichervorganges, der eine Vielzahl von Kommunikationsschnittstellen durchläuft, in und aus einem gemeinsamen Format zum Routen innerhalb des Systems unter der Vielzahl von Kommunikationsschnittstellen, Ressourcen zum Zwischenspeichern der Daten die Gegenstand der Speichervorgänge sind, und Logik zum Verwalten der Migration von Datengruppen von einem Speicherort zu einem anderen Speicherort innerhalb des Netzwerkes.

In einem Ausführungsbeispiel ist das System gemäß der vorliegenden Erfindung in einem Speicherbereichsnetzwerk als ein Zwischengerät enthalten, zwischen Client-Prozessoren, wie zum Beispiel Dateiservern, und Speichersystemen, die als Speicherressourcen in einem Speicherbereich für die Clients verwendet werden.

Speichervorgänge werden von dem Zwischengerät empfangen und gemäß der Konfiguration des Speicherbereiches, der durch die Konfigurationslogik in dem Zwischengerät definiert ist, verwaltet. Das Zwischengerät schafft einen Verwaltungsort innerhalb des Speicherbereichsnetzwerkes, das eine flexible Konfiguration, Redundanz, Failover, Datenmi-

gration, Zwischenspeichern und Unterstützung von zahlreichen Protokollen ermöglicht. Darüber hinaus schafft ein Zwischengerät in einem Ausführungsbeispiel die Emulation von übernommenen Systemen und erlaubt, daß der Speicherbereich ein übernommenes Speichergerät für den Client umfaßt, ohne die Notwendigkeit einer Rekonfiguration des Client. Speicherbereiche werden verwaltet, indem ein logischer Speicherumfang Clients innerhalb des Netzwerkes zugewiesen wird und indem Speicherressourcen in dem Netzwerk auf logische Speicherbereiche der Clients abgebildet werden. Das Zuweisen der logischen Speicherbereiche an Clients wird erreicht, indem in einem Zwischensystem oder einem anderen System, das logisch unabhängig ist oder isoliert ist, der Client der Speicherressourcen in dem Netzwerk auf den logischen Speicherumfang, der dem Client zugewiesen ist, abgebildet wird. Auf diese Weise wird ein Speicherbereich von Speicherressourcen, auf die über einen Speicherbereichsmanager zugegriffen werden kann, verwaltet unter der Verwendung des Speicherbereichsmanagers als einem Zwischengerät.

Ein Speicherserver gemäß der vorliegenden Erfindung umfaßt eine Verarbeitungseinheit, ein Bussystem, das mit der Verarbeitungseinheit verbunden ist, eine Kommunikationsschnittstelle und ein Betriebssystem, das mit der Verarbeitungseinheit verbunden ist. Das Bussystem hat Slots, die geeignet sind, um Schnittstellen für Datenspeicher aufzunehmen, die sich entweder in dem Servergehäuse befinden oder über Kommunikationskanäle mit den Slots verbunden sind. Das Betriebssystem stellt Logik zur Steuerung von Transfers über das Bussystem bereit. Das Betriebssystem stellt Logik für das Übersetzen von Speichervorgängen bereit, die über die Kommunikationsschnittstelle von Client-Servern in einem internen Format empfangen werden. Das Betriebssystem stellt Logik bereit zur Verarbeitung des internen Formats gemäß der Konfigurationsdaten, die einen Speichervorgang auf den Kommunikationsschnittstellen für eine bestimmte Speichereinheit innerhalb des Bereichs des Protokolls des Vorgangs auf eine virtuelle Verbindung abbildet, der diesem Bereich entspricht, unter der Verwendung des internen Formats. Die virtuelle Verbindung wiederum verwaltet das Routen des Vorgangs zu einem oder mehreren physikalischen Datenspeichern durch einen oder mehrere Treiber in den Schnittstellen. Ferner umfaßt der Server Ressourcen zum Emulieren von physikalischen Speichergeräten, so daß Client-Server in der Lage sind, Standardspeichervorgangsprotokolle für den Zugriff auf die virtuellen Geräte ohne Veränderungen in der Konfiguration des Client-Servers für die Speichervorgänge zu verwenden.

Gemäß einem weiteren Aspekt der Erfindung wird ein Speicherrouter geschaffen. Der Speicherrouter umfaßt eine erste Kommunikationsschnittstelle, andere Kommunikationsschnittstellen, eine Verarbeitungseinheit und ein Bussystem. Das Bussystem ist mit der Verarbeitungseinheit, der ersten Kommunikationsschnittstelle und den anderen Kommunikationsschnittstellen verbunden. Die Verarbeitungseinheit unterstützt ein Betriebssystem. Das Betriebssystem leitet Speichervorgänge, die über die erste Kommunikationsschnittstelle empfangen werden, an geeignete andere Kommunikationsschnittstellen gemäß den Konfigurationsdaten weiter unter der Verwendung der virtuellen Gerätearchitektur und der Emulation.

In einigen Ausführungsbeispielen ist die Kommunikationsschnittstelle eine Schnittstelle zu einem faseroptischen Medium. In einigen Ausführungsbeispielen umfaßt die Kommunikationsschnittstelle Treiber, die mit einer Faserkanal vermittelten Schleife kompatibel sind. In einigen Ausführungsbeispielen umfaßt die Kommunikationsschnittstelle Treiber, die mit der Standard "Kleincomputersystem-Schnittstelle Version 3" (small computer system interface version 3, SCSI-3) kompatibel sind.

In einigen Ausführungsbeispielen weist die Verarbeitungseinheit eine Vielzahl von Verarbeitungseinheiten auf. In einigen Ausführungsbeispielen weist das Bussystem verbundene Computerbusse auf. In einigen Ausführungsbeispielen sind die Computerbusse kompatibel mit einem Standard "Verbindungsbus für Umgebungskomponenten (peripheral component interconnect, PCI, Bus). In einigen Ausführungsbeispielen ist die Kommunikationsschnittstelle mit dem Bussystem verbunden.

In einigen Ausführungsbeispielen umfaßt der Speicherserver nichtflüchtigen Speicher. In einigen Ausführungsbeispielen umfaßt der nichtflüchtige Speicher einen integrierten Schaltkreis mit nichtflüchtigem Speicher, wie zum Beispiel einen Flashmemory.

In einigen Ausführungsbeispielen umfaßt der Speicherserver Controller für ein Plattenlaufwerk. In einigen Ausführungsbeispielen unterstützt der Controller ein Feld von Plattenlaufwerken. In einigen Ausführungsbeispielen unterstützt der Controller Standard "Redundanzfelder von unabhängigen Laufwerken (redundant arrays of independent disks, RAID) Protokoll". In einigen Ausführungsbeispielen sind die Plattenlaufwerke mit den Controllern über ein faseroptisches Medium verbunden. In einigen Ausführungsbeispielen haben die Plattenlaufwerke doppelte Schnittstellen zur Verbindung mit einem faseroptischen Medium. In einigen Ausführungsbeispielen ist jedes Plattenlaufwerk mit zumindest zwei Controllern verbunden.

In einigen Ausführungsbeispielen umfaßt das Betriebssystem Logik zum Übersetzen von SCSI-3-Anweisungen und -Daten, die über die Kommunikationsschnittstelle empfangen werden in ein internes Format. In einigen Ausführungsbeispielen wird die logische Einheitsnummer (logical unit number, LUN), die der SCSI-3-Anweisung zugeordnet ist, dazu verwendet, um die SCSI-3-Anweisung und -Daten virtuellen Geräten zuzuweisen, inklusive Datenspeichern in dem Speicherserver. In einigen Ausführungsbeispielen werden die Initiator SCSI-3-Identifizierungsnummer (ID) und die LUN dazu verwendet, um die SCSI-3-Instruktionen und -Daten virtuellen Geräten zuzuordnen, inklusive Datenquellen, die mit dem Speicherserver verbunden sind.

In einigen Ausführungsbeispielen umfaßt das Betriebssystem Logik zur Überwachung der Leistungsfähigkeit und des Zustands des Speicherservers. In einigen Ausführungsbeispielen gibt es Logik zur Behandlung von Ausfällen von Geräten und zum Transfer der Steuerung an redundante Komponenten.

Die vorliegende Erfindung schafft eine Speicherserverarchitektur, die virtuelle Geräte und virtuelle Verbindungen zum Speichern und Verwalten von Daten unterstützt. Der Speicherserver gemäß der vorliegenden Erfindung umfaßt eine Vielzahl von Kommunikationsschnittstellen. Eine erste Gruppe von Kommunikationsschnittstellen in der Vielzahl ist für eine Verbindung zu allen Arten von Anwendern von Daten geeignet. Eine zweite Gruppe von Kommunikationsschnittstellen in der Vielzahl ist für eine Verbindung zu entsprechenden Geräten in einem Pool von Speichergeräten geeignet. Datenverarbeitungsressourcen in dem Speicherserver sind mit der Vielzahl von Kommunikationsschnittstellen verbunden zum Transfer von Daten unter den Schnittstellen. Die datenverarbeitenden Ressourcen umfassen eine Vielzahl von Treiber-

modulen und konfigurierbarer Logik, die Treibermodule in Datenpfade verbindet, die in Paaren implementiert werden für eine Redundanz in einem bevorzugten System. Jeder konfigurierte Datenpfad dient als eine virtuelle Verbindung, die eine Gruppe von Treibermodulen umfaßt, die aus der Vielzahl von Treibermodulen ausgewählt worden sind. Ein Datenspeichervorgang, der an einem Kommunikationsinterface empfangen wird, wird auf einen der konfigurierten Datenpfade abgebildet.

Gemäß einem weiteren Aspekt der Erfindung umfaßt die Vielzahl der Treibermodule einen Protokollserver für ein Protokoll, das auf einer Kommunikationsschnittstelle in der Vielzahl von Kommunikationsschnittstellen unterstützt wird. Der Protokollserver erkennt Zielidentifizierer, die bestimmte Speicherbereiche identifizieren gemäß dem Protokoll auf der Schnittstelle. Vorgänge, die an einen bestimmten Speicherbereich adressiert sind, werden auf einen bestimmten konfigurierten Datenpfad in dem Server abgebildet.

Die Datenpfade, die auf diese Weise konfiguriert sind, dienen als virtuelle Speichergeräte. Die Anwender der Daten kommunizieren mit einer Kommunikationsschnittstelle auf dem Speicherserver gemäß einem Protokoll für ein bestimmtes Speichergerät. Innerhalb des Servers werden die Vorgänge gemäß diesem Protokoll auf ein virtuelles Speichergerät abgebildet, das durch eine Gruppe von Treibern implementiert wird. Das Einrichten und Verändern der Speicheraufgaben, die in einem speziellen Datenpfad durchgeführt werden, und das Einrichten und Verändern der Abbildungen von einem Speicherbereich von einem Datenpfad zu einem anderen, werden durch das Konfigurieren der Gruppe von Treibermodulen innerhalb des Speicherservers erreicht.

Gemäß einem Aspekt der Erfindung umfaßt die Vielzahl der Treibermodule ein oder mehrere Hardwaretreibermodule, die entsprechende Kommunikationsschnittstellen verwalten und ein oder mehrere interne Treibermodule, die unabhängig von der Vielzahl von Kommunikationsschnittstellen die Aufgaben des Datenpfades durchführen. Die Aufgaben des Datenpfades umfassen beispielsweise die Verwaltung des Zwischenspeichers, die Verwaltung des Spiegelns von Speichern, die Verwaltung des Partitionierens von Speichern, die Verwaltung der Datenmigration und anderer Aufgaben zur Verwaltung von Speichervorgängen. Durch das Erfüllen von Datenpfadaufgaben dieser Art in der virtuellen Gerätearchitektur ist die Konfiguration des Speichersystems zur Verwaltung dieser Aufgaben im wesentlichen transparent für die Anwender. Zusätzlich erlaubt das Bereitstellen der virtuellen Gerätefähigkeit bei einem Speicherserver, der für die Durchführung dieser Aufgaben optimiert ist, eine verbesserte Leistungsfähigkeit und eine größere Flexibilität.

Gemäß einem Aspekt der Erfindung umfaßt die Vielzahl der Treibermodule ferner Logik zum Kommunizieren von Daten innerhalb der Serverumgebung gemäß eines internen Nachrichtenformats. Ankommende Speichervorgänge werden in das interne Nachrichtenformat übersetzt und in dem konfigurierten Datenpfad für den jeweiligen Vorgang angeordnet. In einem bevorzugten Ausführungsbeispiel führt der Protokollserver die Übersetzung des Protokolls und die Funktion des Abbildens auf die virtuelle Verbindung durch.

Die konfigurierbare Logik umfaßt ein Anwenderinterface zur Aufnahme von Konfigurationsdaten und einen Speicher, der Tabellen oder Listen der entsprechenden Gruppe von Treibermodulen speichert, die die Datenpfade umfassen.

Die konfigurierbare Logik ist in einem Ausführungsbeispiel implementiert unter der Verwendung einer grafischen Benutzeroberfläche, beispielsweise auf einem Anzeigeschirm inklusive eines Touch Screens zur Aufnahme von Eingangssignalen. Die grafische Anwenderoberfläche ermöglicht die Implementierung von Konfigurationswerkzeugen, die flexibel und leicht zu verwenden sind.

Gemäß einem weiteren Aspekt der Erfindung umfaßt die Konfigurationslogik Speicher zum Speichern von Konfigurationsdaten in der Form von Tabellen, die die Datenpfade für die virtuellen Verbindungen identifizieren.

Der Speicher wird in einem Ausführungsbeispiel implementiert unter der Verwendung eines dauerhaften Tabellenspeicherprozesses, der die Tabellen in einem nichtflüchtigen Speicher hält, der ein Reset und/oder ein Herunterfahren des Speichersystems übersteht. Zusätzlich implementiert die Konfigurationslogik die Datenpfade für die virtuellen Verbindungen unter der Verwendung von redundanten Treibermodulen auf redundanter Hardware in dem System. Daher wird keine einzelne Stelle des Versagens in dem Speichersystem mit einem speziellen Speichervorgang interferieren.

In einem bevorzugten Ausführungsbeispiel sind die Ressourcen innerhalb der Speicherbereiche definiert unter der Verwendung von virtuellen Verbindungen, die eine Vielzahl von Treibermodulen und konfigurierbarer Logik umfassen, die die Treibermodule in Datenpfade verbindet, die zur Redundanz in einem bevorzugten System in Paaren implementiert sind. Jeder konfigurierter Datenpfad arbeitet als eine virtuelle Verbindung, die eine Gruppe von Treibermodulen umfaßt, die aus der Vielzahl von Treibermodulen ausgewählt sind. Ein Datenspeichervorgang, der an einer Kommunikationsschnittstelle empfangen wird, wird auf einen der konfigurierten Datenpfade abgebildet und dadurch innerhalb eines Speicherbereiches gesteuert, der in dem Speicherbereichsmanager verwaltet und konfiguriert wird.

Die Speicherbereichsverwaltung ermöglicht in fundamentaler Weise, daß für Kunden das volle Versprechen von Speicherbereichsnetzwerken zur Behandlung von Geschäftsproblemen Wirklichkeit wird. Die Speicherbereichsverwaltungs-Plattform schafft heterogene Interoperabilität der Speichersysteme und Protokolle, schafft sichere zentralisierte Verwaltung, schafft Skalierbarkeit und hohe Leistungsfähigkeit und schafft Zuverlässigkeit, Verfügbarkeit und Wartungsmerkmale, alles in einer intelligenten, für diesen Zweck gebauten Plattform.

Andere Aspekte und Vorteile der vorliegenden Erfindung kann man bei der Betrachtung der Figuren der detaillierten Beschreibung und der folgenden Ansprüche erkennen.

KURZE BESCHREIBUNG DER FIGUREN

Fig. 1 erläutert ein Speicherbereichsnetzwerk mit einem Speicherserver gemäß der vorliegenden Erfindung, der als ein Speicherrouter oder als ein Speicherverwaltungsdirektor eines Speicherbereichs konfiguriert ist.

Fig. 1A erläutert eine Vielzahl von Anwendungen für intelligente Speicherbereichsnetzwerkserver.

Fig. 2 erläutert ein Speicherbereichsnetzwerk in einer alternativen Konfiguration mit einem Speicherserver gemäß der vorliegenden Erfindung, der als ein Speicherrouter oder als ein Speicherdirektor bei der Verwaltung von Speicherbereichen in einem heterogenen Netzwerk konfiguriert ist.

Fig. 3 erläutert ein komplexeres Speicherbereichsnetzwerk mit mehreren Speicherververn gemäß der vorliegenden Er-

findung mit direkten Kommunikationskanälen zwischen ihnen zur Unterstützung eines erweiterten Speicherbereichs oder Speicherbereichen.

Fig. 4 ist ein Blockdiagramm eines Speicherservers zur Unterstützung der Speicherbereichsverwaltung gemäß der vorliegenden Erfindung.

Fig. 5 ist ein alternatives Diagramm eines Speicherservers zur Unterstützung der Speicherbereichsverwaltung gemäß der vorliegenden Erfindung.

Fig. 6 ist ein Blockdiagramm einer Hardware-Architektur eines intelligenten Speicherbereichsnetzwerkservers.

Fig. 7 ist ein Blockdiagramm der Softwaremodule eines Betriebssystems und von Unterstützungsprogrammen für einen intelligenten Server eines Speicherbereichsnetzwerkes.

Fig. 8 ist ein vereinfachtes Diagramm eines Hardware-Treibermoduls für eine Faserkanalschnittstelle zur Verwendung in dem System der vorliegenden Erfindung.

Fig. 9 ist ein vereinfachtes Diagramm eines Festkörperspeichersystems unter der Verwendung eines Hardware-Treibermoduls der vorliegenden Erfindung.

Fig. 10 ist ein Diagramm eines internen Feldes von Plattenlaufwerken, die in einem Ausführungsbeispiels eines Speicherservers gemäß der vorliegenden Erfindung befestigt sind.

Fig. 11 ist ein vereinfachtes Diagramm eines internen Servicemoduls für einen Zielserver gemäß der vorliegenden Erfindung mit einer lokalen Antwortfähigkeit.

Fig. 12 ist ein Diagramm eines internen Servicemoduls zur Implementierung einer Plattenspiegelung.

Fig. 13 ist ein Diagramm eines internen Servicemoduls zur Implementierung einer Partitionierungsfunktion.

Fig. 14 ist ein Diagramm eines internen Servicemoduls zur Implementierung einer Zwischenspeicherfunktion.

Fig. 15 erläutert eine virtuelle Verbindungskonfiguration gemäß der vorliegenden Erfindung.

Fig. 16 ist ein Diagramm eines internen Servicemoduls zur Implementierung eines dauerhaften Tabellenspeichermanagers gemäß der vorliegenden Erfindung.

Fig. 17 erläutert schematisch ein Hardware-Treibermodul für einen dauerhaften Speicher gemäß der vorliegenden Erfindung.

Fig. 18 ist ein vereinfachtes Diagramm eines Netzwerkes mit einem Zwischengerät mit dreistufigen Hot-Copy-Ressourcen gemäß der vorliegenden Erfindung.

Fig. 19 erläutert Datenstrukturen, die in einem Beispiel eines Treibers zur Implementierung eines Hot-Copy-Vorgangs gemäß der vorliegenden Erfindung verwendet werden.

Fig. 20 ist ein Flußdiagramm, das einen Hot-Copy-Vorgang zeigt, der von einem Treiber gemäß der vorliegenden Erfindung ausgeführt wird.

Fig. 21 ist ein Flußdiagramm, das die Behandlung einer Schreibanforderung während eines Hot-Copy-Vorgangs erläutert.

Fig. 22 ist ein Flußdiagramm, das die Behandlung einer Leseanforderung während eines Hot-Copy-Vorgangs erläutert.

DETAILLIERTE BESCHREIBUNG

Überblick

Fig. 1 erläutert ein Netzwerk inklusive eines intelligenten Speicherbereichsnetzwerkes (intelligent storage area network, ISAN)-Servers **1200**, der eine Speicherbereichsverwaltung bereitstellt. Ein Speicherbereichsnetzwerk (storage area network, SAN) kann dazu verwendet werden um Datenspeicherdienste für Clientcomputer bereitzustellen. Ein Speicherbereichsnetzwerk ist optimiert, um hohe Bandbreiten und hohen Durchsatz von Speicher für Clientcomputer, wie zum Beispiel Dateiserver, Webserver und die Computer von Endanwendern bereitzustellen. Ein Speicherserver **1200** gemäß der vorliegenden Erfindung stellt in bevorzugten Ausführungsbeispielen Datenspeicherplatz im Gehäuse, Zwischenspeicherdienste für Speichervorgänge, Speicherrouten und virtuelle Geräteverwaltung bereit.

Der Speicherserver **1200** in dem Netzwerk hat Clientschnittstellen **1210**, **1211** und **1212**, die mit entsprechenden Clientservern **1201**, **1202** und **1203** verbunden sind. Speicherschnittstellen **1213** und **1214** sind über Kommunikationskanäle mit Speichergeräten **1205**, **1206** und **1207** verbunden, die, wenn sie mit irgendeinem Speicher in dem Speichergerät **1200** verbunden sind, physikalischen Speicher für einen Speicherbereich bereitstellen, der in dem Speicherserver **1200** verwaltet wird.

Der Kommunikationskanal **1213** ist in diesem Beispiel über ein Hub **1204** mit den Geräten **1205** und **1206** verbunden. Beim Betrieb arbeiten die Clientschnittstellen gemäß einem Protokoll, durch das der Clientserver Speichervorgänge durch Befehle anfordert, die Parameter enthalten, die für die Identifizierung eines Speicherbereichs ausreichend sind, inklusive beispielsweise eines oder mehrerer Identifizierer eines Initiators, eines logischen Bereichs, wie zum Beispiel einer LUN-Nummer und eines Identifizierers eines Zielgerätes. Der Speicherserver **1200** bildet die gewünschte Transaktion auf ein virtuelles Gerät ab, das wiederum physikalischen Speicherplatz zur Verwendung in dem Vorgang innerhalb der physikalischen Speichergeräte allokiert. Der Speicherserver **1200** enthält ferner Ressourcen, die die in der Anfrage identifizierten physikalischen Zielgeräte emulieren. Der Speicherserver **1200** ist in der Lage, Speichervorgänge unter der Verwendung von lokalen Konfigurationsdaten weiterzuleiten und die Verwaltung von Speicher für die Clientserver zu vereinfachen.

Um den höchsten Durchsatz zu schaffen ist der Speicherserver **1200** mit den Clientservern **1201–1203** durch ein Hochgeschwindigkeits-Netzwerkmedium, wie zum Beispiel einen Faserkanal oder ein Gigabit-Ethernet verbunden. Die Clientserver **1201–1203** sind in typischen Konfigurationen mit den Computern von Endanwendern durch Netzwerkverbindungen verbunden.

Fig. 1 illustriert eine Verwaltungsschnittstelle **108**, die mit dem Server **1200** über die Kommunikationsverbindung **109** verbunden ist. Die Kommunikationsverbindung, die durch die Schnittstellen in der Station **108** und in dem Server **1200**

bedient wird, umfaßt beispielsweise eine Ethernet-Netzwerkverbindung, ein serielles Kabel, das mit seriellen Ports verbunden ist, oder eine interne Busschnittstelle in verschiedenen Ausführungsbeispielen.

Die Kommunikation zwischen den Servern **1201–1203** und den Speichergeräten **1205–1207** wird durch ein mit einem Glasfaserkanal vermitteltes Schleifennetzwerk bereitgestellt durch den Speicherserver **1200** als ein Zwischengerät. Die Kanäle über das FC-AL können erreicht werden unter der Verwendung eines Protokolls, das mit der Clientcomputer-Systemschnittstelle Version 3 (SCSI-3) kompatibel ist, vorzugsweise unter der Verwendung eines Faserkanalmediums, das auch Faserkanalprotokoll (FCP) bezeichnet wird (beispielsweise $SCSI_BX3T10$ und FCP 10-300.269-199X). In anderen Ausführungsbeispielen werden Protokolle, wie zum Beispiel das Internetprotokoll, über das Faserkanalgefüge zum Transportieren von Speichervorgängen in einer Vielzahl von Protokollen verwendet. In einigen Ausführungsbeispielen unterstützt der Speicherserver **1200** viele Protokolle für die Datenspeichervorgänge.

Fig. 1A erläutert eine Vielzahl von Verwendungen für intelligente Speicherbereichsnetzwerkserver (ISAN-Server). Ein Speicherbereichsnetzwerk (SAN) kann dazu verwendet werden um Datenspeicherdienste für Clientcomputer bereitzustellen. Ein Speicherbereichsnetzwerk ist optimiert zum Bereitstellen von hohen Bandbreiten und hohem Speicherdurchsatz für Clientcomputer, wie zum Beispiel einen Dateiserver oder einen Webserver. Ein ISAN-Server schafft zusätzliche Funktionalitäten über das Datenspeichern und -abrufen hinaus, wie zum Beispiel Speicherrouten und die Verwaltung von virtuellen Geräten.

Fig. 1A umfaßt die Server 100A–D, die ISAN-Server 102A–F, die dünnen Server 104A–C und ein Speicherfeld **106**. Die Server 100A–D können UNIX-Server, Windows™ NT-Server, NetWare™-Server oder irgendein anderer Typ von Dateiserver sein.

Die Server 100A–D sind mit Clientcomputern über Netzwerkverbindungen verbunden. Der ISAN-Server **102A** ist mit dem Server **100A** über eine Netzwerkverbindung verbunden. Der ISAN-Server **102A** stellt Datenspeicherdienste für den Server **100A** bereit durch das Durchführen der gewünschten Speichervorgänge. Der ISAN-Server **102A** wird von dem Server **100A** wie ein Speichergerät behandelt. Der ISAN-Server **102A** ist in der Lage, mehr Speicher zu enthalten als eine typische Festplatte oder ein Feld von Festplatten. Der ISAN-Server **102A** kann verwendet werden als ein Speicherrouter und dazu dienen, intelligentes Routen unter Datenspeichern, die mit dem ISAN-Server **102A** verbunden sind, bereitzustellen.

Der ISAN-Server **102A** stellt ferner höhere Bandbreiten und höheren Durchsatz bei der Verarbeitung von Speichervorgängen bereit, als ein typisches Festplattenlaufwerk oder ein Feld von Festplattenlaufwerken. Der ISAN-Server **102A** kann daher das Volumen von Anfragen behandeln, die erzeugt werden durch Multimediadatenströme und andere großvolumige Datenströme.

Um den höchsten Durchsatz zu schaffen, kann der ISAN-Server **102A** mit dem Server **100A** durch ein Hochgeschwindigkeit-Netzwerkmedium, wie zum Beispiel einen Faserkanal, verbunden werden. Die Server 100B–D sind mit Clientcomputern durch Netzwerkverbindungen verbunden. Die Server 100B–D sind mit einem Speicherbereichsnetzwerk durch ein Faserkanalgerüst verbunden. Das Speicherbereichsnetzwerk umfaßt die ISAN-Server 102B–D und das Speicherfeld **106**. Die Server 100B–D und die ISAN-Server 102B–D unterstützen Treiber für eine Faserkanal vermittelte Schleife (FC-AL).

Kommunikation zwischen den Servern 100B–D und den Speichergeräten über das FC-AL kann erreicht werden unter der Verwendung eines Protokolls, das mit der Standard-Clientcomputersystem-Schnittstelle Version 3 (SCSI-3) kompatibel ist unter Verwendung vorzugsweise eines Fiberkanalmediums das auch eines Fiberkanalprotokoll (FCP) bezeichnet wird (beispielsweise $SCSI_BX3T10$ und FCP X3.269-199X). In anderen Ausführungsbeispielen werden andere Protokolle, wie zum Beispiel das Internetprotokoll, dazu verwendet, über das Fiberkanalgerüst **108** Speichervorgänge in einer Vielzahl von Protokollen zu befördern. In einigen Ausführungsbeispielen unterstützt der ISAN-Server **102A** mehrere Protokolle.

Die dünnen Server 104A–C sind mit den Clients über Netzwerkverbindungen verbunden, verwenden jedoch nicht Speicherbereichsnetzwerke, um Datenspeicher bereitzustellen.

Die ISAN-Server 102E–F sind direkt mit den Clients über Netzwerkverbindungen verbunden. Es gibt keine Zwischendateiserver. Die ISAN-Server 102E–F können applikationsspezifische Prozessoren (ASPs) bereitstellen, die Funktionalitäten wie zum Beispiel Dateiserver, Webserver und andere Typen von Verarbeitung, bereitstellen.

Fig. 2 erläutert ein weiteres Ausführungsbeispiel eines Speicherbereichsnetzwerkes. In **Fig.** 2 ist ein Server **1250**, der Speichersteuerlogik und Zwischenspeicher, wie oben erläutert, enthält, mit Clientservern in einer Vielzahl von verschiedenen Plattformen verbunden, inklusive eines Hewlett-Packard-Servers **1255**, eines Sun-Servers **1256** und eines SGI-Servers **1257**, die jeweils verschiedene Protokolle ausführen zur Verwaltung von Speichervorgängen. Eine Vielzahl von physikalischen Speichergeräten, die die physikalischen Ressourcen zur Verwendung als Speicherbereiche bilden, ist ebenfalls mit dem Server **1250** verbunden, und wird durch den Speicherdirektor gemäß der oben beschriebenen virtuellen Geräte-Architektur verwaltet. Die Vielzahl der physikalischen Speichergeräte umfaßt in diesem Beispiel Speicher auf einer Hewlett-Packard-Plattform **1251**, Speicher auf einer Sun-Plattform **1252** und Speicher auf einer EMC-Plattform **1253**. Daher ermöglicht der Server inklusive der Speichersteuerlogik die Erzeugung eines gemeinsamen Speicherpools, der übernommene Server und Speicher in einer heterogenen Umgebung unterstützen kann. Inkompatibilitäten unter der Vielzahl von Speichergeräten und Servern kann maskiert oder wie gewünscht nachgemacht werden unter der Verwendung der virtuellen Geräte-Architektur. Wahre Speicherbereichsnetzwerkumgebungen können implementiert werden und alle Host-, Gerüst- und Speicher-Interoperabilitätsfragen können auf dem Niveau des Speicherservers verwaltet werden.

Die Speichersteuerlogik schafft unter Verwendung der virtuellen Geräte-Architektur einen einzigen intelligenten Koordinationspunkt für die Konfiguration des Clientserver-Zugriffs auf den Speicher unter Verwendung der Speicherbereichskonfigurationen. Wenig oder keine Hardware-Rekonfiguration ist notwendig beim Hinzufügen neuer Geräte oder dem Verändern der Verwaltung von existierenden Geräten. Die Konfiguration des Speicherservers stellt eine genaue Konfigurationsinformation und Kontrolle bereit, indem sie die automatische Aufrechterhaltung der Abbildung von Datengruppen im physikalischen Speicher auf Servern ermöglicht. Die Aufrechterhaltung von genauen Abbildungen des

physikalischen Speichers vereinfacht signifikant die Verwaltung von Speicherbereichsnetzwerken. Ferner ermöglicht die Speichersteuerung am Server die aktive Migration von Daten von alten Speichergeräten auf neue Speichergeräte, während die Geräte online bleiben. Zusätzlich sind Speicherobjekte in ihrer Größe nicht länger limitiert durch die Größe des größten Objektes, das in einem Feld erzeugt werden kann. Mehrere Felder können zu einem einzigen Speicherobjekt verkettet werden, unabhängig von den Host-Betriebssystemen, die auf den Clientservern laufen. Die Speichersteuerung kann ferner Backup- und Testvorgänge verwalten, wie zum Beispiel das Erzeugen von Schnappschüssen der Daten in dem nichtflüchtigen Speicher und die Verwaltung von Daten-Backups durch das Kopieren der Daten von einer Platte auf ein Band, beispielsweise, ohne durch den Clientserver geroutet zu werden.

Darüber hinaus kann der lokale Zwischenspeicher verwendet werden, um Daten von Feldern, die Redundanz verloren haben, zu verschieben, und um den redundanten Speicher zu reparieren und die volle Verfügbarkeit der Daten zu erhalten, während ein Feld repariert oder wiederaufgebaut wird. Für Anwendungen mit mehreren Servern, die auf eine gemeinsame Gruppe von Daten zugreifen, kann Verschlußlogik in dem Speicherserver in einer Weise angeordnet werden, die eine einfache skalierbare Lösung schafft, unter der Verwendung der virtuellen Geräte-Architektur.

Die Speichersteuerlogik in dem Speicherserver dient zur Zusammenlegung von Zwischenspeicheranforderungen von sowohl den Servern als auch dem Speicher, um die Gesamtmenge von Zwischenspeicher, der für ein Speicherbereichsnetzwerk benötigt wird, zu verringern.

Das System ist in der Lage, entweder für den Clientserver oder das Speichersystem mehr Zwischenspeicher zu allokieren, als einer von beiden effektiv als einen interner Speicher bereitstellen kann. Ferner kann der Zwischenspeicher dynamisch oder statisch allokiert werden, so wie es von der Anwendung, die das System verwendet, definiert wird.

Fig. 3 erläutert ein exakteres Beispiel eines Speicherbereichsnetzwerkes unter der Verwendung einer Vielzahl von verbundenen Speicherservern gemäß der vorliegenden Erfindung. Speicherserver **1300, 1301** und **1302** sind enthalten und verbunden durch Kommunikationskanäle **1350, 1351**, die beispielsweise ein Hochgeschwindigkeitsprotokoll, wie zum Beispiel einen Faserkanal, Gigabit-Ethernet oder asynchronen Transfermodus (Asynchronous Transfer Mode, ATM) verwenden.

In dem bevorzugten Ausführungsbeispiel umfaßt jeder Speicherserver Speichersteuerlogik und nichtflüchtigen Zwischenspeicher. Die Speicherserver **1300, 1301** und **1302** sind in diesem Beispiel mit einer Vielzahl von Clientservern **1310** bis **1318** verbunden. Die Clientserver **1313** und **1314** sind über ein Hub **1320** mit dem Speicherserver **1301** verbunden. In ähnlicher Weise sind die Clientserver **1316** bis **1318** über ein Hub **1321** verbunden, der wiederum mit dem Speicherserver **1302** verbunden ist.

Die Clientserver **1310** bis **1318** kommunizieren mit dem Speicherserver unter der Verwendung von Speicherkanalprotokollen, wie zum Beispiel FCP, das oben genau beschrieben wurde.

Gemäß dieser Protokolle werden Speichervorgänge angefordert und beinhalten einen Identifizierer oder einen Initiator der Anforderung, eine logische Einheitsnummer (logical unit number, LUN) und einen Identifizierer des Zielspeichergerätes. Diese Parameter werden durch die Speichersteuerlogik verwendet, um den Speichervorgang auf ein virtuelles Geräte innerhalb eines Speicherbereiches abzubilden.

Die Server umfassen ferner Ressourcen zur Emulation des Zielspeichergeräts, so daß die Clientserver glatt mit der Vielzahl von Speichergeräten in dem Speicherbereichsnetzwerk zusammenarbeiten.

In **Fig.** 3 gibt es eine Vielzahl von Speichergeräten **1330** bis **1339** die als mit den Speicherservern **1300–1302** verbunden dargestellt sind. In dem Diagramm werden eine Vielzahl von Symbolen verwendet, um die Speichergeräte darzustellen und um anzuzeigen, daß das Netzwerk heterogen ist und ein breites Spektrum von Geräten durch die virtuellen Geräteschnittstellen an den Servern **1301** bis **1302** verwaltet werden. Ferner können die Kommunikationskanäle variiert werden. Daher sind Hubs **1340, 1341** und **1342** in dem Netzwerk enthalten, um eine Vielzahl von Kommunikationsprotokollen zwischen den Speichergeräten und den Speicherservern zu erleichtern.

Ein intelligenter Speicherbereich-Netzwerkserver

Fig. 4 ist ein Blockdiagramm eines Speicherservers in einem bevorzugten Ausführungsbeispiel, der Speichersystem-Verwaltungsressourcen gemäß der vorliegenden Erfindung umfaßt.

Der Speicherserver **102** hat Verbindungsoptionen **130** inklusive einer Gruppe von Kommunikationsschnittstellen, die geeignet sind für Anwender und andere Datenverarbeitungsfunktionen und Speicheroptionen **128** inklusive einer Gruppe von Kommunikationsschnittstellen, die für Speichergeräte geeignet sind. Der Speicherserver **102** hat eine Hardwareschnittstelle **126**, ein Betriebssystem **124**, eine Blockspeicherschnittstelle **118**, eine Verwaltungsschnittstelle **120** und eine Protokollschnittstelle **122**. Die Verbindungsoptionen **130** umfassen serielle Verbindungen **140**, eine Front-Panel-Verbindung **142** zur Unterstützung einer Konfigurationsverwaltungsroutine in einem Ausführungsbeispiel, eine Ethernet-Verbindung **144** zur Unterstützung der Kommunikation mit einer entfernten Verwaltungsstation und ein Netzwerkinterface **146**. Die Speicheroptionen **128** umfassen das Laufwerkfeld **132**, das Festkörperlaufwerk (solid state drive, SSD)-Laufwerk **134**, die SCSI-Schnittstelle **136** und die Netzwerkschnittstelle **138**. Die SCSI-Schnittstelle **136** ist mit einem DVD/CD-R **148** verbunden. Die Netzwerkschnittstelle **138** ist mit einem Speicherserver **102**G und/oder Speicher **150** verbunden.

Die Verbindungsoptionen **130** sind verschiedene Verfahren zum Verbinden von Server und Clients mit dem Speicherserver **102**. Die seriellen Verbindungen **140** unterstützen Netzwerkverwaltung, Modems für eine ferngesteuerte Verwaltung und ununterbrechbare Energieversorgungsnachrichten. Die Front-Panel-Verbindung **142** unterstützt eine Verwaltungsverbindung mit der Front-Panel-Anzeige des Speicherservers **102**. Die Ethernet-Verbindung **144** unterstützt eine Ethernet-Schnittstelle für Verwaltungsprotokolle und möglicherweise für Datentransfer. Die Netzwerkschnittstelle **146** ist eine von möglicherweise vielen Hochgeschwindigkeits-Schnittstellen auf dem Server. In einigen Ausführungsbeispielen ist die Netzwerkschnittstelle **146** eine Faserkanalschnittstelle mit Treibern für eine Faserkanal vermittelte Schleife (fibre channel arbitrated loop, FC-AL). Die Netzwerkschnittstelle **146** kann ferner Treiber für SCSI-3 über das Faserkanalmedium enthalten unter der Verwendung eines Faserkanalprotokolls (fibre channel protocol, FCP).

Die Hardwareschnittstelle **126** stellt schnittstellenspezifische Hardwarekomponenten bereit. Beispielsweise hat die Netzwerkschnittstelle **146** eine für die Netzwerkschnittstelle spezifische Gruppe von Softwaremodule zur Unterstützung von Konfiguration, Diagnose, Leistungsüberwachung und Gesundheits- und Statusüberwachung.

Das Betriebssystem **124**, die Tabellen **116** und die Schnittstellen **118–122** unterstützen die virtuellen Geräte und die Funktionalität des Speicherroutens des Speicherservers **102**. Diese Komponenten des Speicherserver **102** routen Speichervorgänge unter den geeigneten Speicheroptionen **128** und den Verbindungsoptionen **130** unter der Verwendung von konfigurierten Gruppen von Treibermodulen in dem System.

Das Betriebssystem **124** stellt das Routen von Nachrichten und Transportmöglichkeiten zusätzlich zu Sicherungsmöglichkeiten bereit. Das Routen von Nachrichten und die Transportmöglichkeiten des Betriebssystems **124** werden verwendet, um Nachrichten inklusive Speichervorgängen zwischen den Komponenten des Speicherservers **102** zu routen. Diese Nachrichten umfassen Nachrichten in dem internen Format zwischen den Komponenten einer virtuellen Verbindung. Diese Nachrichten können ferner Kontrollnachrichten in anderen Formaten umfassen.

Die Blockspeicherschnittstelle **118** stellt Softwaremodule bereit zur Unterstützung von Blockdatentransfers. Die Schnittstelle **118** umfaßt Unterstützung für gestreifte (striped) Datenspeicherung, gespiegelte Datenspeicherung, partitionierte Datenspeicherung, Zwischenspeicherung, und RAID-Speicherung. Die verschiedenen unterstützten Speichertypen können verbunden werden, um verschiedene Kombinationen, wie zum Beispiel gespiegelte Datenspeicherung mit einem Zwischenspeicher zu bilden.

Die Protokollschnittstelle **122** schafft Softwaremodule zum Übersetzen und Antworten auf Anfragen in einer Vielzahl von Protokollen. Eine Gruppe von Modulen wird bereitgestellt für die Schichten einer Ethernet-Verbindung: der Hardwaretreiber, der Datenverbindungstreiber, der Internetprotokoll (IP)-Treiber der Übertragungskontrollprotokoll (transmission control protocol, TCP)-Treiber, der Anwenderdatagrammprotokoll (user datagramm protocol, UDP)-Treiber und andere Treiber. Eine andere Gruppe von Modulen stellt Treiber für FCP bereit.

Die Verwaltungsschnittstelle **120** schafft Softwaremodule zur Verwaltung des Speicherservers **102**. Die Verwaltungsschnittstelle **120** enthält Schnittstellen zum Verwalten des Zugriffs auf die Tabellen **116**. Die Verwaltungsschnittstelle **120** enthält ferner Schnittstellen für eine regelbasierte Verwaltung des Systems inklusive: des Aufstellens eines Plans oder die Organisation eines Prozesses; die Überwachung des Systems; informiertes Zustimmungsmanagement; und die Behandlung von Systemprozessen und Ereignissen. Das informierte Zustimmungsmanagementmodul basiert auf dem Bereitstellen von regelbasierten Verwaltungsvorschlägen zum Konfigurieren und Warten des Speicherservers **102**.

Das Behandeln von Speichervorgängen

Speichervorgänge werden über eine der Verbindungsoptionen **130** empfangen. Speichervorgänge umfassen Lese- und Schreibanforderungen ebenso wie Statusanfragen. Die Anforderungen können blockorientiert sein.

Ein typischer Lesespeichervorgang umfaßt den Lesebefehl und Adreßinformation. Ein Schreib-Speichervorgang ist ähnlich dem Lesespeichervorgang mit der Ausnahme, daß die Anforderung Information über die Menge an Daten, die gesandt werden, umfaßt, und daß ihr die Daten, die geschrieben werden sollen, folgen. Insbesondere hat bei der Verwendung des SCSI-3-Protokolls jedes Gerät einen Identifizierer (identifier, ID). Die Maschine, die die Anforderung ausgibt, wird der Initiator genannt und die Maschine, die auf die Anfrage antwortet, wird das Ziel genannt. In diesem Beispiel ist der Server **100A** der Initiator und hat einen ID 7. In diesem Beispiel ist der Speicherserver **102** das Ziel und hat einen ID 6. Das SCSI-3-Protokoll stellt zwei oder mehr Adreßkomponenten bereit, eine logische Einheitsnummer (logical unit number, LUN) und eine Adresse.

Die LUN spezifiziert eine Unterkomponente der Ziel-ID. Beispielsweise können sich in einem kombinierten Festplatten/Bandlaufwerkgehäuse die zwei Geräte einen Identifizierer teilen, aber unterschiedliche LUNs haben. Die dritte Adreßkomponente ist die Adresse, von wo die Daten gelesen werden sollen oder wohin sie gespeichert werden sollen. Der Speicherserver **102A** schafft virtuelle LUNs auf einer Initiatorbasis. Daher kann ein einzelner Speicherserver **102A** beispielsweise zehntausend virtuelle LUNs oder mehr unterstützen.

Der Speicherserver **102A** wird die Anforderung des SCSI-3-Speichervorgangs auf eine virtuelle Verbindung abbilden, entsprechend einer virtuellen LUN. Eine virtuelle Verbindung ist eine Folge von einem oder mehreren virtuellen Geräten. Ein virtuelles Gerät besteht aus einem oder mehreren Geräten, wie zum Beispiel einem Softwaremodul oder Hardwarekomponenten. Beispielsweise können zwei Netzwerkschnittstellengeräte kombiniert werden, um ein virtuelles Gerät zu sein. In ähnlicher Weise können zwei Zwischenspeichergeräte kombiniert werden als ein virtuelles Gerät. Dieser Aufbau ermöglicht das virtuelle Komponenten versagen, ohne daß die Fähigkeiten zur Verarbeitung von Speichervorgängen des Speicherservers **102** unterbrochen werden.

Eine virtuelle Verbindung umfaßt die notwendigen virtuellen Geräte zur Unterstützung eines Speichervorgangs. Typischerweise ist die erste Komponente in der virtuellen Verbindung ein Treiber zur Übersetzung des Speichervorgangs vom Format des Kommunikationskanals des Speichervorgangs – FCP in diesem Beispiel – in ein internes Format. Ein solches internes Format kann ähnlich sein dem Nachrichtenformat der intelligenten Eingangs- und Ausgangs (intelligent input and output, I_2O) -Blockspeicherarchitektur (block storage architecture, BSA). Das interne Format ist in dem bevorzugten System neutral in bezug auf das Speichermedium und den Kommunikationskanal.

Das virtuelle Zwischengerät einer virtuellen Verbindung stellt zusätzliche Dienste wie zum Beispiel das Zwischenspeichern, das Spiegeln, RAID, etc. bereit. Da das interne Format neutral ist in bezug auf das Speichermedium, können alle der virtuellen Zwischengeräte ausgelegt sein, auf dem internen Format zu arbeiten und damit mit anderen virtuellen Geräten in der Verbindung zusammenarbeiten.

Das abschließende virtuelle Gerät in einer virtuellen Verbindung ist typischerweise die Formatübersetzung und die Kommunikationskanaltreiber zur Steuerung der Speicherung. Beispielsweise wird ein Laufwerksfeld **132** gesteuert durch redundante Hardwaretreibermodule (redundant hardware driver modules, HDMs) die gruppiert sind, um ein virtuelles Gerät zu bilden. Die HDMs stellen BSA für SCSI-Übersetzung bereit und das HDM behandelt die Schnittstelle zu den Treibern, die das Laufwerkfeld **132** bilden. In ähnlicher Weise wird es ein virtuelles Gerät mit Unterstützung für

BSA-Übersetzung für das Kommunikationskanalprotokoll des Speichergeräts geben, wenn die virtuelle Verbindung eine Verbindung zu einem anderen Typ von Speicher über die Netzwerkschnittstelle **138** ist.

Der Speicherserver umfaßt ferner Ressourcen in dem Betriebssystem und bei den Schnittstellen zu den Clientservern, die physikalische Speichergeräte emulieren. Die Emulation ermöglicht, daß es für die Clientserver beim Zugriff auf den Speicher so erscheint, als ob die virtuellen Geräte physikalische Geräte wären.

Daher können die Clientserver konfiguriert werden unter der Verwendung von Standardprotokollen, wie zum Beispiel FCP, unter der Verwendung von SCSI-Befehlen für Speichervorgänge. In dem Ausführungsbeispiel unter der Verwendung von SCSI-Befehlen bringt die Emulation das Antworten auf einen Anfragebefehl gemäß dem SCSI-Protokoll mit Geräteidentifizierern mit sich und mit Information über die Gerätefähigkeit, die von dem initiierenden Server erwartet wird oder mit ihm kompatibel ist. Auch ein Lesekapazität-Befehl und ein Modepage-Datenbefehl in dem SCSI-Protokoll werden durch die Emulationsressourcen in einer Weise behandelt, die ermöglicht, daß die Clientserver, die den Speicher verwenden, sich auf Standardkonfigurationsinformation für physikalische Speichergeräte verlassen, während der Speicherserver die Clientserver täuscht, indem er die physikalischen Speichergeräte an der Schnittstelle mit dem Clientserver emuliert und die tatsächlichen Speichervorgänge auf virtuelle Geräte abbildet. Die Emulationsressourcen erlauben ferner, daß virtuelle Geräte identifiziert werden durch die Kombination eines Initiators, einer logischen Einheitsnummer (logical unit number, LUN) und eines Identifizierers für ein Zielgerät, ohne daß es notwendig ist, daß der Speichervorgang an das spezifische physikalische Zielgerät, das in den Anforderungen identifiziert ist, gebunden ist.

Fig. 5 ist ein Blockdiagramm, das funktionale Komponenten eines Servers zeigt, wie desjenigen, der mit Bezug auf **Fig. 4** erläutert worden ist und der als ein Speicherverwaltungssystem **151** zur Verwendung bei der Speicherbereichsverwaltung dient. Das System **151** umfaßt ein Speicherverwaltungsbetriebssystem **152**. Mit dem Speicherverwaltungsbetriebssystem **152** umfassen funktionale Komponenten Speicherbereichsroutingressourcen **153**, Ressourcen zur Emulation von übernommenen Geräten **154**, Datenmigrationsressourcen **155** und Redundanz, Hot Swap und Ausfallressourcen **156**. Das Speicherverwaltungsbetriebssystem koordiniert die Kommunikation unter den Ressourcen, einem auf dem Gehäuse angeordneten (on-chassis) Zwischenspeicher **157**, einer Verwaltungsschnittstelle **158** und in diesem Ausführungsbeispiel einem On-Chassis-Speicherfeld **159**.

Der Zwischenspeicher **157** umfaßt ein nichtflüchtiges Festkörperspeicherfeld in einem Ausführungsbeispiel der Erfindung zur sicheren Unterstützung der Speichervorgänge. In einem anderen Ausführungsbeispiel umfaßt der Zwischenspeicher **157** Redundanzfelder für zusätzliche Fehlertoleranz.

Eine Vielzahl von Kommunikationsschnittstellen **160–165** wird auf dem System **151** geschaffen. In diesem Beispiel ist die Schnittstelle **160** geeignet, um das Protokoll X zwischen einem Client und dem Speicherverwaltungssystem **151** auszuführen; die Schnittstelle **161** ist geeignet zur Ausführung des Protokolls Y zwischen einem Client und dem Speicherverwaltungssystem **151**; die Schnittstelle **162** ist geeignet zur Ausführung des Protokolls Z zwischen einem Speichergerät und dem Speicherverwaltungssystem **151**; die Schnittstelle **163** ist geeignet zur Ausführung des Protokolls A zwischen einem Speichergerät und dem Speicherverwaltungssystem **151**; die Schnittstelle **164** ist geeignet zur Ausführung des Protokolls B zwischen einem Speichergerät und einem Speicherverwaltungssystem **151**; und die Schnittstelle **165** ist geeignet zur Ausführung des Protokolls C zwischen dem Speicherverwaltungssystem **151** und einem weiteren Speicherverwaltungssystem auf dem Netzwerk.

In dem erläuterten Beispiel werden die Protokolle X–Z und die Protokolle A–C durch das Speicherverwaltungssystem **151** unterstützt. Diese Protokolle können mehrere unterschiedliche Protokolle sein, Varianten eines einzelnen Protokolls oder alle das gleiche Protokoll, so wie es für ein jeweiliges Speicherbereichsnetzwerk geeignet, ist, indem das System verwendet wird.

Speichervorgänge durchlaufen die Schnittstellen **160–165** von entsprechenden Kommunikationsmedien zu den internen Ressourcen des Speicherverwaltungssystems **151**. In einem bevorzugten System werden Speichervorgänge in ein gemeinsames systeminternes Nachrichtenformat übersetzt zum Routen unter den verschiedenen Schnittstellen, unabhängig von den Protokollen, die durch diese Schnittstellen ausgeführt werden. Ressourcen **153** zum Speicherbereichsrouten bilden die Vorgänge innerhalb des Speicherbereichs ab unter der Verwendung von virtuellen Verbindungen, die für die jeweiligen Clientgeräte und Speichergeräte konfiguriert sind. Ressourcen **154** zur Emulation von übernommenen Geräten und Datenmigrationsressourcen **155** ermöglichen, daß ein Speicherbereich bei dem Speicherverwaltungssystem **151** rekonfiguriert wird, wenn neue Ausrüstung hinzugefügt wird und vom Netzwerk entfernt wird. Beispielsweise kann ein neues Speichergerät zu dem Netzwerk hinzugefügt werden und eine Datengruppe in einem existierenden Speichergerät kann auf ein neues Speichergerät verschoben werden und es kann der Anschein erweckt werden, daß Speichervorgänge von Clients, die die Datengruppe verwenden, erscheinen als ob sie auf den existierenden Speichergeräten verbleiben während der Migration und nachdem die Migration abgeschlossen ist, indem eine Zielemulation bereitgestellt wird. Die Redundanz, die Hot Swap und die Ausfallressourcen **156** gewährleisten eine Fehlertoleranz und unterstützen den kontinuierlichen Betrieb des Speicherverwaltungssystems **151** für Datenspeichernetzwerke mit hohem Durchsatz.

Überblick über die Hardware-Architektur

Fig. 6 ist ein Blockdiagramm einer geeigneten Hardware-Architektur eines (Speicher)-Servers für ein intelligentes Speicherbereichsnetzwerk. Die Hardware-Architektur implementiert Redundanz und unterstützt verteilte Softwaresysteme zum Verhindern, daß ein Versagen an irgendeinem einzelnen Punkt mit einem bestimmten Speichervorgang interferiert.

Fig. 6 umfaßt den Speicherserver **102A**. Der Speicherserver ist ausgelegt, um ein hohes Maß an Redundanz bereitzustellen unter der gleichzeitigen Verwendung von Standardkomponenten und auf einem Standard basierenden Geräten. Beispielsweise verwendet der Speicherserver **102A** eine Hochgeschwindigkeitsversion der Standardumgebungskomponenten-Verbindungsimplementierung (peripheral component interconnect, PCI) und eine Standard Faserkanal vermittelte Schleife (standard fibre channel arbitrated loop, FC-AL) Schnittstelle. Eine Vielzahl von anderen Protokollen und Schnittstellen können in anderen Ausführungsbeispielen verwendet werden.

Der Speicherserver **102**A hat vier separate 64-Bit 66 MHz PCI-Busse 200A–D. Viele unterschiedliche Konfigurationen von Speichergeräten und Netzwerkschnittstellen in den Slots der PCI-Busse sind möglich. In einem Ausführungsbeispiel sind die PCI-Busse in zwei Gruppen aufgeteilt: die SSD PCI-Busse 200A–B und die Schnittstellen PCI-Busse 200C–D. Jede Gruppe hat zwei Busse, die durch die Begriffe oberer und unterer bezeichnet werden. Die oberen und unteren Busse in jeder Gruppe können konfiguriert werden um, Redundanzdienste bereitzustellen. Beispielsweise hat der untere SSD PCI-Bus **200**B die gleiche Konfiguration wie der obere SSD PCI-Bus **200**A.

Die PCI-Busse 200A–D sind mit den Hostbrückencontroller (host bridge controller, HBC)-Modulen 202A–B verbunden.

Die HBC-Module 202A–B überspannen die PCI-Busse 200A–D und stellen redundante Brückenpfade bereit.

Die SSD PCI-Busse 200A–B unterstützen Festkörpertreiber (solid state drive, SSD)-Module 204A–G. Die SSD-Module 204A–G stellen Festkörperspeichergeräte wie zum Beispiel Flashmemoryspeicher bereit.

Die Schnittstellen-PCI-Busse ermöglichen eine Verbindung von den Netzwerk-Schnittstellencontroller (network interface controller, NIC)-Modulen 206A–B, den redundanten Feldern von unabhängigen Laufwerken (redundant arrays of independent disks, RAID)-Controllermodulen (RAC) 212A–B und den Modulen 208A–D zur anwendungsspezifischen Verarbeitung (application specific processing, ASP) mit den HBC-Modulen 202A–B.

Zusätzlich zur Verbindung des Speicherservers **102**A mit dem externen FC-AL können die NICs 206A–B mit dem Faserkanalhub (fibre channel hub, FCH)-Modulen 214A–D verbunden werden. Jedes FCH-Modul 214A–D ist mit beiden NIC-Modulen 206A–B verbunden. Jedes FCH-Modul 214A–D stellt zehn FC-AL-Ports bereit, und kann über die NIC-Module 206A–B kaskadiert werden, um ein FC-AL-Hub mit zwanzig Stationen bereitzustellen.

Die Laufwerkshubmodule (disk drive hub, DDH) 216A–D stellen ein redundantes FC-AL-Gerüst bereit zur Verbindung von Laufwerken mit den RAC-Modulen 212A–B. Das FC-AL-Gerüst umfaßt in jedem der DDH-Module 216A–D zwei redundante Schleifen, die alle Laufwerke, die mit dem DDH-Modul verbunden sind mit beiden RAC-Modulen 212A–B verbindet. Die RAC-Module verwalten eine Schleife unter allen DDH-Modulen 216A–D. Die DDH-Module 216A–D unterstützen jeweils fünf Plattenlaufwerke mit zwei Ports sowie das Plattenlaufwerk **218**.

Die Systemmittelebene (system mid-plane, SMP) ist in **Fig.** 6 nicht dargestellt. Die SMP ist eine passive Mittelebene, die Verbindungen bereitstellt, die in **Fig.** 6 gezeigt sind, zwischen dem HBC-Modul 201A–B, den SSD-Modulen 204A–H, den RAC-Modulen 212A–B, den NIC-Modulen 206A–B, den FCH-Modulen 214A–D, den DDH-Modulen 216A–D und den ASP-Modulen 208A–D. Die SMP basiert auf kompaktem PCI mit vier custom kompakten PCI-Bussen 200A–D, RAC-DDH-Verbindungen und NIC-FCH-Verbindungen und verschiedenen Kontrollbussen, umfassend die Mittelebenensignale. Zusätzlich stellt die SMP Stromverteilung von den Stromsubsystemen (nicht dargestellt in **Fig.** 6) an die Module bereit mit Spannungen von 48 V, 12 V, 5 V und 3,3 V.

Die Front-Panel-Anzeige (panel display, FPD) **220** stellt ein Anwenderinterface für den Speicherserver **102**A bereit. Die FPD enthält ein Anzeigegerät und ein Eingabegerät. In einem Ausführungsbeispiel wird ein berührungssensitiver Flüssigkristallbildschirm (liquid crystal display, LCD) verwendet, um einen berührungsssensitiven Schirm mit Eingabefähigkeiten darzustellen. Die FPD **220** ist mit den HBC-Modulen 202A–B verbunden, um Statusanzeigen, Konfigurationsanzeige und Verwaltung und andere Verwaltungsfunktionen zu unterstützen.

Strom und Belüftungssubsysteme (nicht dargestellt in **Fig.** 6) stellen redundante Wechsel-zu-Gleichstrom-Stromversorgungen dar, redundante Gleichstrom-zu-Gleichstrom-Leistungskonversion, Batteriebackup für Stromausfälle und ein red- undantes Push-Pull-Lüftersubsystem. Diese Komponenten unterstützen die hohe Verfügbarkeit und die Merkmale einer niedrigen Ausfallzeit, die wichtig sind, wenn ein Speicherbereichsnetzwerk verwendet wird. Der Speicherserver **102**A kann mit anderen Speicherservern verbunden werden, um als ein einzelner Netzwerkport in einem Speicherbereichsnetzwerk zu erscheinen oder als ein Netzwerk mit hinzugefügtem Speichergerät. Diese Verbindung kann erzeugt werden über die FC-AL-Expansionsports, die mit jedem der HBC-Module 202A–B verbunden sind. Zusätzlich bieten die HBC-Module 202A–B RS232 serielle Ports und 10/100 Ethernet-Ports für Out-Of-Band-Verwaltung.

Das Bussystem umfaßt alle Busse in dem Speicherserver **102**A. In diesem Beispiel umfaßt das Bussystem die vier PCI-Busse, die durch die Hostbrückencontroller miteinander verbunden sind. Das Bussystem umfaßt ferner die PCI-Busse innerhalb der HBC-Module, die zusätzliche Schnittstellen bereitstellen. Die Slots umfassen alle Positionen auf dem Bussystem, die Schnittstellen empfangen können. In diesem Beispiel kann jeder der vier PCI-Busse außerhalb der HBC-Module vier Schnittstellen aufnehmen.

Die Schnittstellen sind Karten oder andere Geräte, die in den Slots angeordnet werden. Die Schnittstellen unterstützen Treiber und Hardware für die Datenspeicher, die mit den Schnittstellen verbunden sind.

Redundanz und Fail-Over

Der Speicherserver **102**A bietet ein hohes Maß an Redundanz. In einem Ausführungsbeispiel gibt es redundante NIC-, RAC- und HBC-Module. Die SSD-Module und Laufwerke unterstützen Spiegel. Die Laufwerke unterstützen ferner Parität und Zweikanalzugriff. Jedes DDH-Modul enthält ein vollredundantes FC-AL-Gerüst zur Verbindung mit den RAC-Modulen. Ausfälle werden durch die HBC-Module behandelt, die die anderen Module in den Speicherserver steuern. Die Steuerung besteht aus mehreren Schichten.

Die erste Schicht des HBC-Moduls der Steuerung ist die Steuerung der Stromversorgung. Jedes Modul hat ein individuelles Stromversorgungs-Enablesignal, das durch den CMB-Controller auf dem Modul gesteuert wird. Obwohl die HBC-Module redundant sind, dient nur ein HBC-Modul als das Master-HBC-Modul und steuert und leitet das System. Die anderen HBC-Module dienten als ein Slave.

Wenn ein Modul in einen Slot gesteckt wird, ist seine Stromversorgung anfänglich ausgeschaltet. Nur das HBC-Mastermodul kann die Stromversorgung einschalten. Wenn ein Modul anfängt, inkorrekt zu arbeiten und auf Befehle nicht antwortet, kann das HBC-Modul die Stromversorgung zu dem Modul ausschalten.

Die zweite Schicht der Steuerung für die HBC-Module ist der Card-Management-Bus (CMB). Jedes Modul hat einen Atmel AT90S8515 (AVR) Mikrocontroller, der mit dem CMB verbunden ist. Das HBC-Modul selbst hat einen AVR-Mi-

krocontroller, der mit dem CMB verbunden ist und der als ein Master oder als ein Slave dienen kann. Der CMB-Mikrocontroller wird durch eine Verbindung zu der Mittelebene versorgt, unabhängig von der Leistung, die an den Hauptprozessor auf dem Modul geliefert wird. Das CMB ermöglicht, daß der Master-HBC einen Kartentyp liest, feststellt, ob eine Karte anwesend ist, einen nichtmaskierbaren Interrupt an eine Karte sendet oder einen Hardreset einer Karte durchführt. Modulprozessoren und die Master-HBC-Module können ferner Kommunikation über einen seriellen Port auf dem AVR-Mikrocontroller auf dem Modul durchführen. Dieser Kommunikationspfad kann verwendet werden als ein Backup für Kontrollkommunikation für den Fall eines PCI-Ausfalls.

Die dritte Ebene der Steuerung für die HBC-Module ist der PCI-Bus. Wenn ein Modul nicht antwortet, kann es über den CMB abgefragt werden unter der Verwendung eines Kontrollprozesses auf dem PCI-Bus. Wenn das Modul immer noch nicht antwortet, kann über den CMB ein nichtmaskierbarer Interrupt gesetzt werden. Wenn das Modul immer noch nicht antwortet, kann es über den CMB zurückgesetzt werden. Wenn das Modul nach dem Reset immer noch nicht antwortet, kann es heruntergefahren werden und eine Warnung kann ausgegeben werden, das Modul zu ersetzen.

HBC-Modulredundanz

Die HBC-Modulredundanz und die Ausfallsicherheit unterstützen die Systemredundanz. Obwohl die HBC-Module 202A–B beide gleichzeitig aktiv sein können wird nur eines als der Master durch das HOST_SEL-Signal bezeichnet. Das Master-HBC-Modul stellt eine PCI-Busvermittlung für alle PCI-Busse bereit, steuert alle Leistungsenables für die anderen Module und ist der anerkannte Master auf dem CMB-Gerät. Die PCI-Busvermittlungssignale des Backup-HBC-Moduls und die Leistungsenables werden von dem HOST_SEL-Signal außer Kraft gesetzt. Der CMB wird bei jedem Slave CMB der Karten oder FCB-Gerät durch das HOST_SEL-Signal geschaltet. Das HOST_SEL-Signal wird über einen Widerstand auf die Systemmittelebene (system mid-plane, SMP) mitgenommen, wodurch verursacht wird, daß das HBC-Modul **202A** der Default-Master ist. Das HBC-Modul **202B** kann das HOST_SEL-Signal erzeugen um sich selbst zum Master zu machen, dies wird jedoch typischerweise nur auftreten während Ausfällen oder eines Startes, wenn das HBC-Modul **202A** nicht anwesend ist.

Um die Wahrscheinlichkeit eines Fehlers zu reduzieren, treibt der EVC das HOST_SEL-Signal und verlangt ein Schreiben auf zwei separate Speicherorte eines spezifischen Musters. Dies kann verhindern, daß ein HBC-Modul mit einer Fehlfunktion sich selbst zum Master macht. Die Stromenablesignale beider HBC-Module werden auf die SMP mitgenommen, damit beim Start der Strom für beide Karten eingeschaltet wird. Das HBC-Modul **202A** hat die Kontrolle über die Stromeinschaltung für das HBC-Modul **202B**. In ähnlicher Weise hat das HBC-Modul **202B** die Kontrolle über die Stromeinschaltung für das HBC-Modul **202A**. Um wiederum die Wahrscheinlichkeit eines Fehlers zu reduzieren, verlangt das Setzen eines Stromeinschaltsignals eines HBC-Moduls ein Schreiben auf zwei separate Speicherorte eines spezifischen Musters.

PCI-Brücken unterstützen nicht zwei Hosts. Durch spezielles Konfigurieren der PCI-Brücken können beide HBC-Module so konfiguriert sein, daß sie auf den System-PCI-Bussen sind. Die PCI-Brücken auf beiden HBC-Modulen werden so konfiguriert, daß der Adreßraum, der von einem HBC-Modul gesteuert wird, als ein Speicherplatz betrachtet wird, der als lokal für alle System-PCI-Busse auf den anderen PCI-Brücken des HBC-Moduls abgebildet ist. Fehler können auftreten wenn ein HBC-Modul versucht, vom oder in den PCI-Adreßraum des anderen zu lesen beziehungsweise zu schreiben. Der Fehler wird auftreten, da vier Brücken zu den System-PCI-Bussen den Vorgang bestätigen, wodurch ernsthafte Fehler erzeugt werden. Daher sollte ein HBC-Modul nicht versuchen, auf das andere HBC-Modul über die Systembusse zuzugreifen.

Obwohl die HBC-Module nicht über die PCI-Busse kommunizieren sollten, haben die HBC-Module zwei separate Kommunikationspfade: einen besonderen seriellen Port und den CMB. Der besondere serielle Port ist der primäre Pfad für Kommunikation um zu erlauben, daß Nachrichten weitergeleitet werden um eine Zustandsüberprüfung auf den anderen HBC-Modulen bereitzustellen. Wenn ein serieller Port ausfällt, kann der CMB als ein Backup verwendet werden, um festzustellen welches HBC-Modul ausgefallen ist.

Die HBC-Modulstartreihenfolge

Da beide HBC-Module durch das EVC eingeschaltet werden, wenn das System angeschaltet wird, müssen sie feststellen, ob ein weiteres HBC-Modul vorhanden ist, wenn sie eingeschaltet werden. Dies erfolgt über den CMB. Wenn das HBC-Modul **202A** vorhanden ist, wird es nach Voreinstellung der Master. Wenn das HBC-Modul **202A** beim Anschalten feststellt, daß kein HBC-Modul **202B** vorhanden ist, kann es den Strom zu dem Kartenslot des HBC-Moduls **202B** abschalten. Dies ermöglicht, daß ein zweites HBC-Modul hinzugefügt wird und unter der Steuerung des Master-HBC-Moduls angeschaltet wird. Wenn das HBC-Modul **202A** beim Start feststellt, daß das HBC-Modul **202B** vorhanden ist, sollte es eine Kommunikation über den seriellen Port herstellen. Wenn das HBC-Modul **202B** feststellt, daß das HBC-Modul **202A** nicht vorhanden ist, sollte es sich selbst zum Master-HBC-Modul machen durch das Setzen des HOST_SEL-Signals und den Strom zu dem Kartenslot des HBC-Moduls **202A** abschalten. Wenn das HBC-Modul **202B** feststellt, daß das HBC-Modul **202A** vorhanden ist, sollte es auf ein HBC 0 warten zum Herstellen einer Kommunikation über den seriellen Port. Wenn nach einer bestimmten Zeit die Kommunikation nicht hergestellt worden ist, sollte das HBC-Modul **202B** eine Ausfallsequenz beginnen.

Die HBC-Modulausfallsequenz

Die HBC-Module sollten miteinander in spezifischen Intervallen über die serielle Schnittstelle kommunizieren. Wenn das Backup HBC die serielle Kommunikation mit dem Master-HBC verliert, sollte es versuchen, eine Kommunikation mit dem Master-HBC-Modul über sein CMB herzustellen. Wenn die Kommunikation über den CMB hergestellt werden kann und beide Hosts in Ordnung sind, ist die serielle Kommunikationsverbindung schlecht. Beide Karten sollten Dia-

gnose durchführen um festzustellen, wo sich der Fehler befindet. Wenn sich der Fehler auf dem Backup-HBC-Modul befindet oder nicht isoliert werden kann, sollte ein Alarm ausgelöst werden. Wenn sich der Fehler auf dem Master-HBC-Modul befindet oder eine CMB-Kommunikation nicht hergestellt werden kann, sollte das Backup-HBC-Modul das Master-HBC-Modul ausschalten und sich selbst zum Master machen.

Überblick über die Software-Architektur

Ein Speicherserver wird durch ein Betriebssystem unterstützt, das ausgelegt ist, um die einzigartig hohe Bandbreite, den hohen Durchsatz und die Anforderungen eines Speicherservers zu unterstützen. Das Betriebssystem plant und steuert Datentransfers über die Bussysteme und verwaltet das System. Obwohl eine Anzahl von verschiedenen Betriebssystemen und Softwarekomponentenstrukturen möglich sind, wird in einem Ausführungsbeispiel ein hochmodulares Betriebssystem, das für einen Speicherserver ausgelegt ist, verwendet.

Fig. 7 ist ein Blockdiagramm der Softwaremodule eines Betriebssystems und von Unterstützungsprogrammen für einen Speicherserver.

Fig. 7 umfaßt die folgenden Betriebssystemkomponenten: das Hardware-Schnittstellenmodul **900**, das Nucleus PLUS™-Realzeit-Kernelmodul **902**, das bei Accelerated Technologies, Inc. Mobile, Alabama erhältlich ist, das ISOS-Protokollverwaltungsmodul **904** und das Speicherservicemodul **906**. Das Hardware-Schnittstellenmodul ermöglicht, daß Softwarekomponenten des Speicherservers mit den Hardwarekomponenten des Speicherservers kommunizieren.

Das Nucleus PLUS™-Realzeit-Kernelmodul **902** wird verwendet um grundlegende Betriebssystemfunktionen bereitzustellen wie zum Beispiel: Aufgaben, Reihenfolgen, Signale, Timer und die Unterstützung kritischer Abschnitte. Das Nucleus PLUS™-Realzeit-Kernelmodul **902** wird zu den Softwaremodulen des Speicherservers als Funktionen in C++-Klassen durch das Speicherdienstmodul **906** exportiert.

Das ISOS-Modul **904** ermöglicht, daß der Speicherserver eine Nachrichten-Architektur für Eingabe und Ausgabe unterstützt. Die Hardwaremodule, wie zum Beispiel die Module für RAID-Controller (RAC), die Module für Netzwerk-Schnittstellencontroller (NIC), die Module für ein Festkörperlaufwerk (solid state drive, SSD), die Module für ein Laufwerkhub (disk drive hub, DDH) und die Module für das Fiberkanalhub (fibre channel hub, FCH) sind alle Eingabe/Ausgabe-Prozessoren (input/output processors, IOPs). Das Modul des Masterhost-Brückenprozessors (host bridge processor, HBC) dient als der Host.

Das Speicherservicemodul **906** implementiert Nachrichtenklassen zur Unterstützung des zuverlässigen Transports von Nachrichten zwischen Komponenten. Das Speicherservicemodul **906** unterstützt den Betrieb von Gerätetreibermodulen und die Unterstützung von virtuellen Geräten. Die Gerätetreibermodule (device driver modules, DDMs) und die virtuellen Geräte (virtual devices, VDs) sind die Aufbaublöcke des Speicherserver-Speichersystems. Das Speicherservicemodul **906** ist um die Bereitstellung von Unterstützung herum für Anforderungen für Speichervorgänge organisiert.

In einigen Anwendungen wird ein einzelner Speicherserver, wie zum Beispiel der Speicherserver **102A**, mehrere hundert DDMs aufweisen, die in Verbindung mit den Betriebssystemmodulen **900–906** arbeiten zur Unterstützung von Antworten auf Speicherserveranforderungen. Andere Anwendungen verwenden einige wenige DDMs in verschiedenen Kombinationen.

Softwarekomponenten werden als Gerätetreibermodule (DDMs) implementiert. Ein DDM, das primär Anfragen nach einem Hardwaregerät bedient, wird als ein Hardwaretreibermodul (hardware driver module, HDM) bezeichnet. Ein DDM, das als ein internes Zwischenprogramm dient, wird als ein Zwischenservicemodul (intermediate service module, ISM) bezeichnet. Beispielsweise werden die DDMs die die SSD-Module bedienen, als HDMs bezeichnet. Die DDMs, die Zwischenspeicherdienste, Spiegelungsdienste und andere Typen von Diensten bereitstellen, die nicht direkt mit einem Hardwaregerät verbunden sind, könnten als ISMs bezeichnet werden.

Ein einzelnes DDM kann mehrere Instanzen auf einem einzelnen Speicherserver haben. Beispielsweise gibt es in **Fig. 7** vier Instanzen des Leistungs-, Gesundheits- und Status-PHS-Monitors 908A–D, einen für jedes der vier großen Softwaresubsysteme: das NIC **910**, das RAC **920**, das HBC **930** und das SSD **940**. Jedes DDM hat seine eigene Nachrichtenwarteschlange und eine eindeutige Identifizierung. Beispielsweise kann der PHS-Monitor **908A** auf dem NIC **910** die Geräte-ID (device ID, DID) 0 sein. Jedes DDM listet ferner die Klasse von Speicheranforderungen, die von dem DDM behandelt werden auf, und Betriebssystemmodule routen die Anforderungen zu den DDMs auf der Basis der Klasse der Speicheranforderungen. Anforderungen können geroutet werden durch Anforderungscodes oder durch virtuelle Gerätenummern.

Das NIC-Softwaresubsystem **910** umfaßt drei DDMs: eine Prozessorunterstützung HDM **912A**, eine Eingabe/Ausgabeübersetzung ISM **914A** und den PHS-Monitor **908A**. Das RAC-Softwaresubsystem **920** umfaßt drei DDMs: eine Prozessorunterstützung HDM **912B**, ein Eingabe/Ausgabeübersetzung ISM **914B** und einen PHS-Monitor **908B**. Das HBC-Softwaresubsystem **930** umfaßt: eine Prozessorunterstützung HDM **912C**, eine Eingabe/Ausgabeübersetzung ISM **914C**, eine Kartenverwaltung HDM **916**, ein Systemmonitor DDM **918**, ein Internetprotokoll DDM **921**, ein Front-Panel-Anzeige DDM **922**, eine anwendungsspezifische Prozessorunterstützung DDM **924** und einen PHS-Monitor **908C**. Das SSD-Softwaresubsystem **926** umfaßt ein Festkörperlaufwerkmanagement HDM **926** und einen PHS-Monitor **908D**. Die Front-Panel-Anzeige **950** unterstützt einen Hypertext-Markup-Language (HTML)-Client **928**.

Die **Fig. 8–10** erläutern eine Vielzahl von Hardwaretreibermodulen (HDMs) und die **Fig. 11–14** erläutern eine Vielzahl von internen Zwischenservicemodulen (ISMs) gemäß der bevorzugten Architektur der vorliegenden Erfindung. **Fig. 15** stellt ein vereinfachtes Diagramm einer Gruppe von Treibermodulen bereit, die in Datenpfade konfiguriert worden sind, um als virtuelle Verbindungen zu dienen.

Fig. 8 erläutert eine Netzwerkschnittstellenkarte **520** mit einem HDM **524**. Die Karte **520** hat eine physikalische Schnittstelle **521** zu einem Faserkanalnetzwerk. Ein Netzwerkschnittstellenchip **522**, in diesem Beispiel ein Qlogic-Gerät, wie zum Beispiel ein ISP **2200A** von der Firma Qlogic Corporation aus Costa Mesa, Kalifornien, ist mit der physikalischen Schnittstelle **521** verbunden. Der Netzwerkschnittstellenchip **522** erzeugt Kommunikation, die durch die Linie **523** dargestellt wird, die in dem HDM **524** verarbeitet wird. Das HDM **504** bereitet die Kommunikationen zur Verwen-

dung durch andere Treibermodule in dem System auf. Daher hat die Kommunikation, die durch die Linie **525** dargestellt wird, ein SCSI-Format. Die Kommunikation, die durch die Linie **526** dargestellt wird, hat ein Nachrichtenformat, wie zum Beispiel ein BSA-Format. Die Kommunikation, die durch die Linie **527** dargestellt wird, hat ein Internetprotokoll (IP)-Format. Das HDM ist eine Instanz einer Treiberklasse mit dem Namen "Qlogictreiber" in dem Diagramm und ihm ist in diesem Beispiel die Geräteidentifizierung DID **401** gegeben worden. Die physikalische Schnittstelle wird als NIC#1 identifiziert.

Fig. 9 erläutert ein Speichergerät **720**, das implementiert wird durch ein Feld aus nichtflüchtigen integrierten Verbindungspeichergeräten. Das HDM **722** ist mit dem Feld **721** verbunden und überträgt die Kommunikationen der Blockspeicher-Architektur auf der Linie **723** in ein Format zum Speichern und Abrufen aus dem Feld **721**. In diesem Beispiel wird dem HDM **722** eine Geräteidentifizierung **1130** gegeben. Die physikalische Schnittstelle wird als SSD#4 identifiziert.

Fig. 10 erläutert die Konfiguration eines Feldes **820** von Laufwerken, die an dem Speicherservergehäuse in einer faserkanalvermittelten Schleifen-Architektur in dem bevorzugten Ausführungsbeispiel, das in **Fig.** 6 gezeigt ist, befestigt sind. Das Faserkanallaufwerkhub #0 **216A**, das Kanalplattenhub #1 **216B**, das Faserkanallaufwerkhub #2 **216C** und das Faserkanallaufwerkhub #3 **216D**, die ebenfalls in **Fig.** 6 gezeigt sind, sind verbunden mit den redundanten Hubcontroll-HDMs **821** und **822**.

Die HDMs **821** und **822** sind mit physikalischen faserkanalvermittelten Schleifenverbindungen **823** bzw. **824** verbunden. Dem HDM **821** ist die Geräteidentifizierung **1612** gegeben worden und dem HDM **822** die Geräteidentifizierung **1613**. Die Verbindung **823** ist mit einer Faserkanalschnittstelle **825** verbunden. Die Schnittstelle **825** umfaßt einen Netzwerkschnittstellenchip **826**, der mit einer physikalischen Schnittstelle **840** und mit einem HDM **827** verbunden ist. Ein ISM **828** ist mit dem HDM **827** verbunden und mit dem internen Kommunikationspfad **829**. Der ISM **808** überträgt die Blockspeicher-Architekturkommunikationen auf der Leitung **829** in IOCB-Kommunikationen für das HDM **827**. Das HDM **827** kommuniziert mit dem Netzwerkschnittstellenchip **826**, der wiederum den Faserkanal **823** treibt. Den ISM **828** ist die Geräteidentifizierung **1210** gegeben worden und dem HDM **827** die Geräteidentifizierung **1110**. Die physikalische Schnittstelle **825** wird mit RAC #0 bezeichnet.

Die Faserkanalverbindung **824** ist mit der Schnittstelle **830** verbunden. Die Schnittstelle **830** hat eine Konfiguration wie die Schnittstelle **825**. Daher umfaßt die Schnittstelle **830** eine physikalische Faserkanalschnittstelle **831**, die durch einen Netzwerkschnittstellenchip **832** getrieben wird. Der Netzwerkschnittstellenchip **832** kommuniziert auf dem Kanal, der durch die Linie **833** dargestellt wird mit dem HDM **834**. Das HDM **834** kommuniziert mit ISM **835** über den Kanal **816**. Das ISM **835** verwaltet eine Schnittstelle zu den BSA-Formatnachrichten auf dem Kanal **837**. In diesem Beispiel wird dem ISM **835** die Geräteidentifizierung **1211** gegeben. Dem HDM **834** wird die Geräteidentifizierung **1111** gegeben. Die Schnittstelle **830** wird identifiziert als RAC #1.

Die **Fig.** 11–14 erläutern eine Vielzahl von ISM-Beispielen gemäß der vorliegenden Erfindung, die in Datenpfade konfiguriert werden können.

Fig. 11 zeigt einen SCSI-Zielserver **550**, der ein Beispiel eines Protokollservermoduls gemäß der vorliegenden Erfindung darstellt. Ähnliche Protokollservermodule können implementiert werden für irgendein spezielles Speicherkanal- oder Netzwerkprotokoll, das von Anwendern der Daten, die durch den Speicherserver der vorliegenden Erfindung verwaltet werden, implementiert wird. Der Zielserver **550** hat eine Nachrichtenschnittstelle **551**, die hereinkommende Nachrichten von einem HDM empfängt, beispielsweise dem HDM aus **Fig.** 8, das mit einer Kommunikationsschnittstelle verbunden ist, die für die Verbindung mit einem Anwender geeignet ist. In diesem Beispiel haben die Nachrichten auf der Schnittstelle **551** ein SCSI-Format. In anderen Beispielen können die Nachrichten bereits die BSA-Architektur oder irgendeine andere Architektur haben, die geeignet ist für das Protokoll auf der Kommunikationsschnittstelle, die gerade bedient wird. Der Server **550** umfaßt eine Schalterfunktion **550**, die hereinkommende Nachrichten zu einem SCSI zum BSA-Übersetzer **553** überträgt oder zu einer lokalen Antwortfunktion **554**. Typischerweise werden die Nachrichten von dem Übersetzer **553** als herausgehende Nachrichten auf der Leitung **555** weitergeleitet. Hereinkommende Nachrichten auf der Leitung **555** werden an den Übersetzer **556** geliefert, der die hereinkommenden BSA-Nachrichten in das SCSI-Format, das auf der Leitung **551** verwendet wird, überträgt.

In vielen Fällen kann das SCSI-Zielgerät auf die SCSI-Nachrichten antworten, ohne die Nachricht weiterzurouten unter der Verwendung des lokalen Anwortservice **554**. Viele Statusnachrichten, die sich nicht auf das Lesen oder Schreiben vom Speicher selbst beziehen, werden durch den lokalen Antwortservice **554** behandelt.

Der Zielserver **550** ist in diesem Beispiel eine Instanz einer Klasse SCSI-Zielserver und ihm ist eine Geräteidentifizierung **500** gegeben. Eine Funktion des Protokollservers, beispielsweise des SCSI-Servers **550**, besteht darin, das Ausmaß an Speicher zu identifizieren, das Gegenstand eines Speichervorgangs auf der zugeordneten Schnittstelle ist. Der Speicherbereich wird auf eine virtuelle Verbindung abgebildet unter der Verwendung der konfigurierbaren Logik in dem Speicherserver, wie weiter unten detaillierter beschrieben wird.

Fig. 12 erläutert ein ISM **650**, das eine Datenpfadaufgabe zur Spiegelungsverwaltung durchführt. Das ISM **650** umfaßt eine Schnittstelle **651**, die mit den internen Kommunikationskanälen auf dem Gerät verbunden ist. Logikprozesse **652** empfangen die hereinkommenden Kommunikationen und Daten und verwalten eine Spiegelungsfunktion. Die Logik **652** kommuniziert mit einer Vielzahl von Laufwerkschnittstellen inklusive dem primären Laufwerk **653**, dem sekundären Laufwerk **654**, dem tertiärem Laufwerk **655** und einem Standby-Laufwerk **656**. Obwohl in dem Diagramm ein 3-Wege Spiegel gezeigt ist, kann irgendeinen Anzahl von Spiegelungspfaden implementiert werden für "n-Wege"-Spiegel unter der Verwendung von virtuellen Verbindungen. Obwohl der Begriff "Laufwerkschnittstelle" verwendet wird, können andere Typen von Speichergeräten bei den Spiegelungsfunktionen verwendet werden. Die Laufwerkschnittstellen **653–656** kommunizieren unter der Verwendung von internen Kommunikationskanälen mit den HDM-Modulen, die den Zielspeichergeräten, die bei der Spiegelungsfunktion verwendet werden, zugeordnet sind oder mit anderen ISM-Modulen, wie es für die jeweilige virtuelle Verbindung geeignet ist. In diesem Beispiel wird der Spiegel ISM **650** implementiert als eine Instanz einer Klasse "Spiegel" und ihm wird eine Geräteidentifizierung 10200 gegeben.

Fig. 13 erläutert ein Partitions ISM **750**. Das Partitions ISM **750** umfaßt eine Schnittstelle **751**, die interne Kommunikationen von den anderen Treibermodulen empfängt und eine Schnittstelle **752**, die ebenfalls mit anderen Treibermodu-

len kommuniziert. Das ISM **750** umfaßt Logikprozesse **753**, Datenstrukturen zum Speichern einer Basisadresse **754** und einer Begrenzungsadresse **755** und eine Laufwerksschnittstelle **756**. Der Partitionslogikprozeß **753** konfiguriert das betroffene Speichergerät, das durch den Laufwerksprozeß **756** identifiziert wird unter der Verwendung einer logischen Partitionsfunktion, die nützlich ist für eine Vielzahl von Speicherverwaltungstechniken, so daß das physikalische Gerät als mehr als ein logisches Gerät in den virtuellen Verbindungen erscheint. In diesem Beispiel ist das Partitions-ISM **750** eine Instanz einer Klasse "Partition" und ihm ist eine Geräteidentifizierung 10400 gegeben worden.

Fig. 14 erläutert ein Zwischenspeicher-ISM **850**. Das Zwischenspeicher ISM **850** umfaßt Logikprozesse **853**, die mit einer Schnittstelle **851** zu der internen Nachrichtenweiterleitungsstruktur auf dem Speicherserver kommunizieren. Datenstrukturen in dem Zwischenspeicher ISM **850** umfassen eine lokale Zwischenspeicherzuordnung **854**, eine Zwischenspeichertabelle **855**, die die Daten, die in dem Zwischenspeicher **854** gespeichert sind, identifiziert, und eine Laufwerkschnittstelle **856**. Die Laufwerkschnittstelle kommuniziert auf einem Kanal **857** mit einem HDM, das die jeweilige virtuelle Verbindung, die durch den Zwischenspeicher bedient wird, zugeordnet ist. Der Zwischenspeicher **854** wird in einem Ausführungsbeispiel lokal in dem Speicherserver verwaltet. In einem alternativen Ausführungsbeispiel kann der Zwischenspeicher in einem Hochgeschwindigkeits-, nichtflüchtigen Speicher gespeichert werden, wie zum Beispiel ein Festkörperspeichermodul mit einer Architektur, wie sie in bezug auf **Fig.** 9 beschrieben worden ist. In dem bevorzugten Ausführungsbeispiel ist das Zwischenspeichermodul ISM **850** implementiert als eine Instanz einer Klasse "Zwischenspeicher" und ihm ist eine Geräteidentifizierung 10300 gegeben worden.

Fig. 15 stellt ein heuristisches Diagramm von redundanten virtuellen Verbindungen bereit, die durch die Datenpfade inklusive einer Vielzahl von Treibermodulen gemäß der vorliegenden Erfindung implementiert sind. Virtuelle Verbindungen umfassen eine externe Schnittstelle zur Kommunikation mit einem Anwender der Daten, einen Protokollübersetzer zum Übersetzen von Kommunikationen mit dem Anwender in das Kommunikationsformat der Treibermodule und ein Speicherobjekt, das eine Kommunikationsschnittstelle zu einem Speichergerät umfaßt. Speicheroperatoren, die Datenpfadaufgaben durchführen, können zwischen dem Übersetzer und dem Speicherobjekt existieren. Das optimale Ordnen der Treibermodule, die als Speicheroperatoren, wie zum Beispiel Zwischenspeicher, Spiegelung, Partition etc. dienen, wird durch den Systemkonstrukteur durchgeführt unter der Verwendung der konfigurierbaren Logik, die vom Speicherserver bereitgestellt wird.

In dem Beispiel, das in **Fig.** 15 erläutert ist, wird die externe Schnittstelle durch das NIC #0 bereitgestellt und sein zugeordnetes HDM wird dargestellt durch den Block **1010**. Der Protokollübersetzer wird bereitgestellt durch den SCSI-Zielserver ISM **1011**. Eine Zwischenspeicherfunktion wird durch das ISM **1012** bereitgestellt. Eine Spiegelfunktion wird durch das ISM **1013** bereitgestellt. Auf die Speicherobjekte wird von der Spiegelfunktion **1013** zugegriffen, und sie bestehen aus einer Gruppe von physikalischen Schnittstellen, die in diesem Beispiel ausgewählt sind aus der grundlegenden Faserkanal-Daisychain-Schnittstelle und ihrem zugeordneten HDM, das dargestellt wird durch den Block **1014** oder einer externen LUN-Schnittstelle, den Laufwerken in der faserkanalvermittelten Schleife, auf die über das ISM/HDM-Paar, das durch den Block **1015** und den redundanten Block **1016** dargestellt wird, zugegriffen wird, das Festkörperspeichergerät und sein zugeordnetes HDM, das durch den Block **1017** dargestellt wird und die Schnittstelle zu einem externen Laufwerk und seinem zugeordnetem ISM/HDM-Paar, das durch den Block **1018** dargestellt wird. Separate HDM-Module auf den Faserkanalschnittstellen zu den Laufwerken (01), (02), (03), und (04) verwalten die Kommunikation über die faserkanalvermittelten Schleifen mit den Schnittstellen **1015** und **1016**.

In dem gezeigten Ausführungsbeispiel greift das Spiegelungsmodul **1013** auf die Laufwerke (01), (02) und (04) als primäre, sekundäre, bzw. Standby-Laufwerke für die Spiegelfunktionen zu. Obwohl das Spiegelungsmodul, das in **Fig.** 12 gezeigt ist, eine tertiäre Laufwerkschnittstelle umfaßt, wird dieses tertiäre Laufwerk in dem Beispielsystem nicht benutzt.

Ferner sind in dem Diagramm Partitions ISM-Module **1020** und **1021** gezeigt, die nicht mit den Datenpfaden der gezeigten virtuellen Verbindung verbunden sind. Diese Blocks sind vorhanden um zu erläutern, daß bei der Verwendung der virtuellen Verbindungstruktur neue Module wie Partitionierung zu dem Pfad hinzugefügt werden können durch einfaches Konfigurieren des Speicherservers.

Ein redundanter Datenpfad wird implementiert unter der Verwendung der Schnittstelle NIC #1 und seinem zugeordneten HDM, das durch den Block **1025** dargestellt wird, des SCSI-Zielserver-ISM, das durch den Block **1026** dargestellt wird, das Zwischenspeicher-ISM, das durch den Block **1027** dargestellt wird, und das Spiegelungs-ISM, das durch den Block **1028** dargestellt wird. Redundanz wird in den Datenspeichergeräten erreicht durch die Verwendung der Spiegelungsfunktion. Die redundanten Treibermodule werden in einem bevorzugten Ausführungsbeispiel auf separaten IOPs innerhalb des Speicherservers verteilt.

Wie in **Fig.** 15 erläutert ist, umfaßt jedes der Treibermodule eine eindeutige Treiberidentifizierung, die in den Klammern in den Blöcken aus **Fig.** 15 gezeigt ist. Die eindeutigen Geräteidentifizierungen werden verwendet, um die Konfigurationslogik zu unterstützen, die auf Tabellen in einer Konfigurationsdatenbank basiert, die von dem Speicherserver verwaltet wird, und durch lokale konfigurierbare Logik in dem Speicherserver gesteuert wird.

In dem bevorzugten System werden die Konfigurationstabellen verwaltet durch einen dauerhaften Tabellentreiber, wie zum Beispiel den, der in den **Fig.** 16 und 17 erläutert ist. Unter erneuter Bezugnahme auf **Fig.** 4 speichert der Speicherserver **102** Verwaltungs- und Routinginformation in Tabellen, wie zum Beispiel in Tabellen **116**. Auf die Tabellen **116** kann zugegriffen werden über das Verwaltungsinterface **120**. Die Tabellen **116** werden typischerweise in dauerhaftem Speicher, wie zum Beispiel nichtflüchtigem Speicher, gespeichert werden. Die Tabellen **116** können redundant gehalten werden, um Unterstützung zur Ausfallsicherheit bereitzustellen.

Fig. 16 erläutert ein dauerhaftes Tabellenmodul **1400**, das als eine Instanz einer Klasse "dauerhafte Tabelle" implementiert wird und der grundlegenden Architektur der Treibermodulstruktur folgt. Das dauerhafte Tabellenmodul **1400** umfaßt einen logischen Tabellenzugriffsprozessor **1401** und eine Vielzahl von Unterstützungsfunktionen inklusive eines Tabellendatenzugriffsmanagers **1402**, eines dauerhaften Imagemanagers **1403** und eines dauerhaften Synchronisationsmoduls **1404** für die Tabelleninstanz. Der Tabellendatenzugriffsmanager **1402** ist mit einem Tabellenklassenmanager **1405** in diesem Ausführungsbeispiel verbunden. Der Tabellenklassenmanager verwaltet eine Vielzahl von Konfigurati-

onstabellen inklusive einer ID-Tabelle **1406** für einen Faserkanalport, einer LUN-Exporttabelle **1407**, einer Konfigurationstemplatetabelle **1408**, einer DDM-Roll-Call-Tabelle **1409**, einer virtuellen Gerätetabelle **1410**, einer Speicher-Roll-Call-Tabelle **1411**, einer Faserkanallaufwerk-Roll-Call-Tabelle **1412**, einer externen LUN-Tabelle **1413** und einer Festkörperspeichertabelle **1414**. Die spezielle Konfiguration der Gruppen von Tabellen, die durch das dauerhafte Tabellenmodul **1400** verwaltet werden, kann verändert werden, um auf die jeweilige Implementierung angepaßt zu werden und optimiert zu werden für bestimmte Klassen von Geräten.

Der dauerhafte Imagemanager **1403** und der Synchronisationsmanager **1404** für die Tabelleninstanz kommunizieren mit dem dauerhaften Datenspeichertreiber **1420**, wie in **Fig.** 11 dargestellt, und mit einem zweiten dauerhaften Speichertreiber, der nicht gezeigt ist. Der dauerhafte Datenspeichertreiber **1420** wird implementiert als ein HDM, das eine Instanz einer Klasse "dauerhafter Speicher" ist und ihm wird eine Geräteidentifizierung, folgend dem Modell der oben beschriebenen Treibermodule, gegeben. In dem bevorzugten System kommuniziert das HDM **1420** für den dauerhaften Datenspeicher mit dem Festkörperspeichergerät in dem Speicherserver und stellt schnellen Zugriff auf die Daten, die in den virtuellen Verbindungen verwendet werden, bereit.

In dem dauerhaften Datenspeicher wird eine große Variation von Konfigurationsinformation für das System gehalten. Die DDM-Roll-Call-Tabelle **1409** umfaßt eine Liste aller Instanzen der Gerätetreibermodule und ihrer eindeutigen Geräte-IDs. Die Speicher-Roll-Call-Tabelle **1411** umfaßt eine Liste aller aktiven Speichergeräte, die von dem Speicherserver detektiert werden. Die Roll-Call-Tabellen können verwendet werden durch die virtuelle Gerätetabelle **1410** und durch die Konfigurationswerkzeuge, um virtuelle Verbindungen zu erzeugen. Die LUN-Exporttabelle **1407** stellt eine Technik zum Abbilden der identifizierten Speicherbereiche innerhalb eines Speicherkanalvorgangs auf virtuelle Verbindungen bereit. Die externe LUN-Tabelle **1413** identifiziert logische Speichereinheiten, die in anderen Speicherservern gehalten werden, die über externe Speicherschnittstellen auf dem Speicherserver verbunden sind.

Zwei primäre Tabellen unterstützen das Exportieren von Speicher zu Clients und die Speicherroutingfunktionalität des Speicherservers **102A**. Diese Tabellen sind die Exporttabelle **1407** und die virtuelle Gerätekonfigurationstabelle **1410**.

Die Exporttabelle 1407

Die Exporttabelle **1407** bildet Adreßinformation, die mit einem Speichervorgang empfangen wird, auf eine virtuelle Verbindung oder eine Speicheroption ab. Im Fall von SCSI-3 über eine Faserkanalschnittstelle ist die verwendete Adreßinformation die Initiator ID, die Ziel-LUN und die Zieladresse.

Es ist nicht notwendig, alle diese Information zu verwenden zur Auflösung einer Anforderung, da viele LUNs Initiatoren oder Clients gemeinsam sein können und viele LUNs werden die Zieladresse, beispielsweise den Offset auf dem Speichergerät zur Adressierung innerhalb der virtuellen Verbindung verwenden anstelle der Auswahl unterschiedlicher Verbindungen. Daher ist in einem typischen Ausführungsbeispiel die Exporttabelle **1407** organisiert, wie in Tabelle 1 gezeigt.

Tabelle 1

Protokoll	Protokollspezifische Adressierung (LUN)	Initiatorspezifisch? wenn ja, ID	Erstes virtuelles Gerät im der Verbindung	Primärer Verbindungsinhaber
SCSI	0	Nein	11	NIC0
SCSI	1	Ja, ID=6	30	NIC0
SCSI	1	Ja, ID=5	60	NIC1
SCSI	2	Nein	12	NIC0
TCP/IP	Port2000	Nein	70	NIC0

Die Exporttabelle **1407** kann andere Spalten enthalten, wie zum Beispiel den gegenwärtigen Status der virtuellen Verbindung, die Kapazität der virtuellen Verbindung und andere Information. In einem Ausführungsbeispiel listet die Exporttabelle **1407** die gesamten virtuellen Verbindungen in einer Spalte der Exporttabelle auf.

Tabelle 1 zeigt, daß protokollspezifische Adreßinformation verwendet werden kann, um die Anforderung zu der geeigneten virtuellen Verbindung zu routen. Daher werden nur TCP-Sitzungen, die den Port **2000** als Identifizierer des Zielbereiches des Speichers verwenden, zu der virtuellen Verbindung geroutet, die mit dem virtuellen Gerät mit einer Identifizierung **70** beginnt.

Tabelle 1 zeigt, daß eine einzelne LUN für ein Protokoll mit verschiedenen Geräten verbunden werden kann, abhängig vom Initiator des Speichervorgangs. In diesem Beispiel wird LUN 1 auf verschiedene virtuelle Verbindungen abgebildet, basierend auf der Initiatoridee. Ferner können virtuelle Verbindungen abgebildet werden auf der Basis von anderen Typen von Identifizierern, wie zum Beispiel den weltweiten Namen (World Wide Name, WWN).

Eine beispielhafte Exporttabelle hat die folgende Struktur:

```
#define EXPORT_TABLE "Export_Table"

struct Export Table Entry {

    rowID               ridThisRow;          //ReihenID dieser Tabellenreihe.
    U32                 version;             //Version des Eintrags der Exporttabelle.
    U32                 size;                //Größe des Eintrags der Exporttabelle in Bytes.
    CTProtocolType      ProtocolType;        // FCP, IP, andere.
    U32                 CircuitNumber;       //LUN oder andere.
    VDN                 vdNext;              //Erste virtuelle Gerätenummer in dem Pfad.
    VDN                 vdLegacyBsa;         //Virtuelle Gerätenummer des übernommenen
                                             BSA.
    VDN                 vdLegacyScsi;        //Virtuelle Gerätenummer des übernommenen
                                             SCSI.
    U32                 ExportedLUN;         //Exportierte LUN-Nummer.
    U32                 InitiatorId;         //HostID
    U32                 TargetId             //Unsere ID.
    U32                 FCInstance;          //FC-Schleifennummer.
    String32            SerialNumber;        //Verwendung eines Stringfeldes für eine serielle
                                             Nummer.
    long long           Capacity;            //Kapazität dieser virtuellen Verbindung.
    U32                 FailState;
    U32                 PrimaryFCTargetOwner;
    U32                 SecondaryFCTargetOwner;
    CTReadyState        ReadyState;          //Gegenwärtiger Status.
    CTReadyState        DesiredReadyState;   //Gewünschter Bereitschaftsstatus.
    String16            WWNName;             //Weltweiter Name (64 oder 128-Bit IEEE-
                                             Registriert)
    String32            Name;                //Name der virtuellen Verbindung
```

Die virtuelle Gerätekonfigurationstabelle

Die virtuelle Gerätekonfigurationstabelle verbindet virtuelle Geräte mit den Gerättreibern, die das virtuelle Gerät unterstützen. Die virtuellen Geräte sind zur Unterstützung eines redundanten Aufbaus ausgelegt. Daher bildet die Tabelle für die virtuelle Gerätekonfigurationen virtuelle Gerätenummern auf Gerätemodule ab. In einem Ausführungsbeispiel wird eine Tabelle, wie zum Beispiel Tabelle 2, dazu verwendet, um virtuelle Geräte auf unterstützende Gerätetreiber abzubilden. **Fig.** 15 erläutert die virtuelle Verbindung, die durch Tabelle 2 implementiert wird und mit dem virtuellen Gerät **12** beginnt.

Tabelle 2

Virtuelles Gerät	Primär	Alternativen	Parameter	Status	Klasse
1	4000	4001	N/A	Primär	Dauerhafte Tabelle
10	1210	1211	SO(00)	Alternative	FC-Laufwerk
11	500	501	VD(10)	Primär	SCSI-Ziel
12	500	501	VD(13)	Primär	SCSI-Ziel
13	10300	10301	VD(14)	Primär	Zwischenspeicher
14	10200	10201	VD(15, 16, null, 17)	Primär	Spiegel
15	1210	1211	SO(02)	Primär	FC-Laufwerk
16	1210	1211	SO(03)	Primär	FC-Laufwerk
17	1210	1211	SO(04)	Primär	FC-Laufwerk

Wie Tabelle 2 zeigt, wird für jedes virtuelle Gerät Information über primäre und alternative Treibermodule, die das virtuelle Gerät unterstützen, bereitgestellt. Beispielsweise wird im zweiten Eintrag in Tabelle 2 ein Faserkanallaufwerk auf das virtuelle Gerät (VD) **10** abgebildet.

Das virtuelle Gerät umfaßt das eine oder mehrere Software- oder Hardwaremodule zur Unterstützung des virtuellen Gerätes. Die Parameterspalte wird verwendet, um Initialisierungsinformation bereitzustellen. Im Fall von VD(**10**) ist der Parameter SO(00), was für Speicheroption 0 steht. Jede Gerätetreibermodulklasse hat klassenspezifische Parameter. Speicheroptionstreiber verwenden Parameter zum Spezifizieren einer jeweiligen Speichereinheit. Zwischenspeicherklassen, wie zum Beispiel der Spiegelungstreiber und der Zwischenspeichertreiber verwenden Parameter, die das nächste virtuelle Gerät in der virtuellen Verbindung spezifizieren. Dieses Format ermöglicht, daß ein einzelnes Gerätetreibermodul mehrere Geräte basierend auf den Parametereinstellungen unterstützt. Es ist zu bemerken, daß in Tabelle 2 der Gerätetreiber **1210** durch die virtuellen Geräte **10, 15, 16** und **17** verwendet wird, jedoch jedes Gerät einen unterschiedlichen Parameter für den Treiber spezifiziert.

Die Statusspalte gibt den Status der Software oder Hardwaremodule an, die das virtuelle Gerät unterstützen. Beispielsweise ist im ersten Eintrag in Tabelle 2 der Status "primär" was bedeutet, daß der primäre Gerätetreiber, in diesem Fall **4000**, gerade verwendet wird. Im zweiten Eintrag in Tabelle 2 ist der Status "alternativ" was bedeutet, daß der primäre Gerätetreiber ausgefallen ist oder nicht korrekt antwortet. In diesem Fall wird der alternative Treiber **1211** für den zweiten Eintrag in Tabelle 2 verwendet. Wenn ein Gerät mehr als eine Alternative hat, wird die Statusspalte den gerade verwendeten Treiber anzeigen.

Beispiel

Als ein Beispiel wird ein Speichervorgang betrachtet, der über eine der Verbindungsoptionen **130** zu dem Speicherserver **102**A unter Verwendung des SCSI-Protokolls kommt und der in der Adressierungsinformation als LUN2 bezeichnet ist. Ferner sei angenommen, daß der Speicherserver **102**A konfiguriert ist, wie in den Tabellen 1 und 2 für dieses Beispiel gezeigt.

Die Verbindungsoption, wie zum Beispiel die Netzwerkschnittstelle **146** über die der Speichervorgang empfangen wird, ist mit einem Hardwaregerätetreiber verbunden. Der Hardwaregerätetreiber empfängt den Speichervorgang und schickt ihn abhängig vom Protokoll an ein passendes virtuelles Gerät zur Behandlung des Protokolls.

Beispielsweise werden SCSI-Speichervorgänge an einen Gerätetreiber in der SCSI-Zielklasse gesandt. In ähnlicher Weise werden IP-Speichervorgänge an einen Gerätetreiber in der IP-Zielklasse gesandt. Hier wird der Speichervorgang durchgeführt unter der Verwendung des SCSI-Kommunikationsprotokolls und er wird daher an ein SCSI-Zielgerätetreiber (DID500) geroutet.

Der SCSI-Zielgerätetreiber analysiert ferner die Anforderung. Der erste Teil der Analyse dient dazu festzustellen, auf welche virtuelle Verbindung die Anforderung abgebildet werden soll. Diese Feststellung kann getroffen werden unter der Verwendung der Information in der Exporttabelle. In diesem Beispiel zeigt Tabelle 1 an, daß eine Anforderung, die das SCSI-Protokoll verwendet und LUN 2 spezifiziert, zu der virtuellen Verbindung geroutet werden sollte, die mit dem virtuellen Gerät **12** beginnt. In einem Ausführungsbeispiel werden alle SCSI-Zielanforderungen an denselben SCSI-Zieltreiber für eine einzelne Schnittstelle geroutet. In diesem Ausführungsbeispiel wird die Parameterinformation für das Ziel VD12 verwendet, um das Verhalten des SCSI-Zielgerätes zu steuern, statt daß die Nachricht an ein zweites virtuelles Gerät für ein SCSI-Ziel geroutet wird.

Das SCSI-Zielgerät, hier mit der Treibernummer **500**, übersetzt die SCSI-Nachricht in ein internes Format. Solch ein Format basiert auf dem I_2O-Blockspeicher-Architektur (BSA)-Format. Dieses Format ist neutral in Hinsicht auf das Gerät und das Protokoll und kann durch Zwischengerätetreiber verwendet werden. Sobald die Anforderung in einem internen Format ist, wird sie an das nächste virtuelle Gerät in die virtuelle Verbindung gesandt, wie durch das Parameterfeld angegeben. Im vorliegenden Fall ist der Parameter VD(**13**) oder das virtuelle Gerät **13**.

Die Nachricht wird hier an den VD **13** geroutet, der redundante Zwischenspeichertreiber bereitstellt, hier die Treiber mit den Nummern 10300 und 10301. Die Treiber zum Zwischenspeichern verwenden einen Speicher zum Zwischenspeichern von Speichervorgängen. Basierend auf dem Zwischenspeicheralgorithmus, der von dem Treiber verwendet wird,

wird der Treiber die Speichervorgänge zum nächsten virtuellen Gerät in der virtuellen Verbindung bei geeigneten Intervallen routen. Hier wird das nächste Gerät durch den Parameter VD(14) oder virtuelles Gerät **14** angezeigt.

In dem internen Format wird die Nachricht zu VD 14 geroutet. Das virtuelle Gerät **14** umfaßt redundante Spiegelungstreiber. In diesem Fall werden die Treiber 10200 und 10201 verwendet. Die Spiegelungstreiber implementieren einen Spiegelungsalgorithmus, um ein gespiegeltes Bild des Speichers auf verschiedenen Volumina zu halten. Dieser Spiegelungstreiber unterstützt einen primäre, sekundären und tertiären Speicher ebenso wie einen Standby-Speicher. Andere Spiegelungstreiber können unterschiedliche Algorithmen unterstützen. Dieser Spiegelungstreiber unterstützt ferner das Verbinden eines neuen Speichers, der allmählich synchronisiert wird, mit einem existierenden Speicher. Basierend auf dem von den Treiber verwendeten Spiegelungsalgorithmus und dem Status des gespiegelten Speichers, wird der Treiber Speichervorgänge zu geeigneten virtuellen Geräten in die virtuelle Verbindung routen. Unter der Annahme, daß sowohl der primäre als auch der alternative Speicher funktioniert, wird der Spiegelungstreiber diese Anforderung zu den primären und sekundären Speichern nur gemäß der Parameter VD(15, 16, null, 17) oder den virtuellen Geräten **15** und **16** routen. Die Null in der Parameterliste zeigt an, daß kein tertiäres Laufwerk gegenwärtig für dieses virtuelle Gerät verwendet wird.

Der Spiegelungstreiber kann die Nachrichten des Speichervorgangs seriell oder parallel zu den beiden Geräten routen. In diesem Beispiel wird das Weiterleiten der Nachricht zum virtuellen Gerät **15** betrachtet werden, obwohl das Beispiel ebenfalls erweitert werden kann auf den zweiten Speicher, das virtuelle Gerät **16**. Das virtuelle Gerät **15** umfaßt redundante Treiber zum Steuern eines Faserkanallaufwerks. Die Treiber übersetzen das interne Format in ein Format, das von den Laufwerken verwendet wird, beispielsweise BSA zu SCSI. Die Treiber stellen ferner die Adreßinformation für das Laufwerk bereit. Hier wird der Parameter SO(02) verwendet, um eine Speicheroption auszuwählen, hier das Faserkanallaufwerk mit der Nummer 2.

Dementsprechend wird innerhalb der Speicherplattform auf Hardwarefunktionen (wie zum Beispiel ein Laufwerk oder einen Flashspeicher) und Softwarefunktionen (wie zum Beispiel einen RAID-Streifen oder -Spiegel) immer über Softwaretreiber zugegriffen, die üblicherweise als Geräte bezeichnet werden.

Diese Geräte sind paarweise angeordnet (wobei jedes Mitglied des Paars vorzugsweise zur Redundanz ein separates Bord betreibt) und werden virtuelle Geräte genannt. Diese virtuellen Geräte werden miteinander in verschiedenen Konfigurationen verkettet. Beispielsweise kann ein Spiegelungsgerät zu zwei oder drei Laufwerksgeräten verkettet werden. Durch diese Art der Konfiguration werden Ketten von virtuellen Geräten erzeugt. Diese virtuellen Geräteketten können ergänzt werden, solange sie in irgendein BSA-Typgerät konfiguriert sind, das selbst wiederum in irgendeiner anderen Konfiguration verwendet werden kann.

Virtuelle Geräteketten werden mit einem FCP/SCSI-Zielservergerät verbunden und werden in der LUN-Exporttabelle des FCP-Ziel"Treiber" für den "Export" (d. h. die Möglichkeit des Zugriffs von der Außenwelt über das FCP-Protokoll) abgebildet. An dieser Stelle wird die virtuelle Gerätekette mit einem SCSI-Zielservergerät an ihrem Kopf eine virtuelle Verbindung genannt.

Die Software des virtuellen Verbindungsmanagers, die verantwortlich ist für das Erzeugen von virtuellen Verbindungen, fügt den SCSI-Zielserver"Kopf" zu einer virtuellen Gerätekette hinzu und exportiert daraufhin die virtuelle Verbindung durch das Aktualisieren der Exporttabellen des FCP-Ziels. Die Software unterstützt ferner Löschen, Ruhigstellen und Ausfallvorgänge.

Die Software des virtuellen Verbindungsmanagersà ist ferner verantwortlich für das Erhalten der virtuellen Verbindungstabellen, VCTs, die an einer einzigen Stelle alle virtuellen Geräte in der virtuellen Verbindung auflistet. Diese Information wird benötigt zum Implementieren von vielen Systemvorgängen, wie zum Beispiel Fehlerbehandlung, Hot-Swap und das Herunterfahren.

Wenn sie initialisiert ist, definiert die virtuelle Verbindungs-Managersoftware die VCT selbst in dem dauerhaften Tabellenspeicher. Die virtuelle Verbindungs-Managersoftware achtet ferner auf Einfügungen, Löschungen und irgendwelche Modifikationen an der VCT.

Um eine neue virtuelle Verbindung zu erzeugen, muß die Information die notwendig ist, um eine Instanz eines SCSI-Zielservers zu erzeugen und zum Abbilden und Exportieren der neuen LUN in einem Eintrag in dem VCT angeordnet werden.

Der virtuelle Verbindungsmanager achtet auf Einfügungen in die VCT und wird beim Empfang einer Antwort die folgenden Handlungen durchführen:

1. Versuch zum Validieren der Information in dem neu eingefügten Eintrag. Wenn der Eintrag ungültige Information enthält, wird sein Statusfeld so gesetzt, daß er den Fehler anzeigt und keine weitere Handlung wird durchgeführt.
2. Erzeugen eines neuen SCSI-Zielservergerätes für das LUN der virtuellen Verbindung, die durch den neu eingefügten Eintrag spezifiziert wird.
3. Setzen des Status in dem neuen Eintrag auf "Instantiiere".
4. Der Speicher, der der virtuellen Verbindung zugeordnet ist, wird in der Speicher-Roll-Call-Tabelle als verwendet markiert.
5. Die Exporttabelle wird aktualisiert, um die LUN an den neuen SCSI-Zielserver zu senden.

Wenn ein Eintrag in der virtuellen Verbindung gelöscht wird, wird der virtuelle Verbindungsmanager die folgenden Handlungen vornehmen:

1. Stillegen der virtuellen Verbindung, wenn dies nicht bereits erfolgt ist und Markieren der Verbindung als stillgelegt.
2. Entfernen der Versanddaten der virtuellen Verbindung aus der Exporttabelle.
3. Markieren des Roll-Call-Eintrags, der in dem virtuellen Verbindungseintrag angegeben ist, als nicht verwendet.

4. Deinstantiieren des SCSI-Targetservers, der der virtuellen Verbindung zugeordnet ist.

Der virtuelle Verbindungsmanager achtet ferner auf Modifizierungen an dem "Exportiert"-Feld in der VCT. Wenn das "Exportiert"-Feld in irgendeinem Eintrag in der VCT auf wahr gesetzt wird, wird der virtuelle Verbindungsmanager die folgenden Handlungen vornehmen:

1. Exportieren der virtuellen Verbindung, indem die notwendigen Modifikationen an der Exporttabelle des FCP-Ziels durchgeführt werden.
2. Falls während des Exportvorgangs kein Fehler auftritt, wird das Statusfeld in dem VC-Eintrag gesetzt und das "Exportiert"-Feld wird in einem korrekten Zustand gelassen. Wenn die virtuelle Verbindung nicht exportiert worden ist, wird die Exportiert-Flag auf falsch gesetzt.

Der virtuelle Verbindungsmanager achtet auf Modifikationen an dem "Stillgelegt"-Feld in der virtuellen Verbindungstabelle. Wenn das "Stillgelegt"-Feld in irgendeinem Eintrag in der VCT auf wahr gesetzt wird, führt der virtuelle Verbindungsmanager die folgenden Handlungen durch:

1. Wenn der VC gegenwärtig exportiert ist, wird er nicht mehr exportiert und seine "Exportiert"-Flag wird auf falsch gesetzt.
2. An alle virtuellen Geräte in der virtuellen Verbindung werden Nachrichten zum Stillegen gesendet.
3. Falls irgendein Fehler während des Stillegebetriebs auftritt, wird das Statusfeld in dem VC-Eintrag gesetzt und das "Stillgelegt"-Feld wird in einen korrekten Zustand gelassen, d. h. wenn die virtuelle Verbindung nicht stillgelegt worden ist, wird die Stillgelelgt-Flag auf falsch gesetzt.

Anwenderschnittstelle

Eine Anwender-Schnittstelle kann durch Datenverarbeitungsstrukturen zur Anzeige und zur Verwendung bei der Konfiguration eines Speicherservers gemäß der vorliegenden Erfindung hergestellt werden. Das Bild umfaßt ein Fenster mit einem Feld zur Anzeige eines Logos, ein Feld zur Anzeige von grundlegender Information in bezug auf das Gehäuse des Servers und eine Gruppe von Icons, die, wenn sie ausgewählt werden Verwaltungsanwendungen starten. Routinen, die bereitgestellt sind zur Verwaltung von Hardware und Software, Routinen zur Verwaltung des Anwenderzugriffs und Routinen zur Beobachtung von lang andauernden Prozessen auf den Server werden durch die Buttons gestartet. Gemäß der vorliegenden Erfindung wird eine Funktion zum Definieren von Hosts, die mit dem Server verbunden sind, eine Funktion zum Abbilden von exportierten LUNs auf verwaltete Ressourcen und eine Funktion zur Konfiguration des verwalteten Speichers durch die Buttons gestartet.

Das Fenster enthält ferner eine Anwender-Logon-Dialogbox inklusive eines Feldes zur Angabe eines Anwendernamens und eines Feldes zur Eingabe eines Passwords.

Host-Manager

Der Anwender startet einen Host-Manager unter der Verwendung eines Buttons. Dieser Abschnitt beschreibt ein Java-basiertes Anwender-Interface (UI) zur Definierung von Hosts (Servern) für einen Speicherserver. Die Verwaltungssoftware öffnet ein Fenster, das eine Tabelle präsentiert, mit Einträgen, die einen Hostnamen, eine Portnummer, eine Initiatoridee und eine Beschreibung in mehreren Spalten für jeden Host enthält, der zur Konfiguration und zur Verwendung zur Verfügung steht. Andere Felder umfassen einen Netzwerkschnittstellen-Kartenidentifizierer und einen eindeutigen Hostidentifizierer in anderen Spalten. Der eindeutige Hostidentifizierer ist in dem bevorzugten Beispiel der weltweite Nummernwert für einen Faserkanal-Host.

Der Hostmanager ist eine Subkomponente der Java-basierten Verwaltungsanwendung des Speicherservers, die den Anwender in die Lage versetzt, einem NIC-Port und einer Initiator-ID einen Namen und eine Beschreibung zuzuordnen, um den Prozeß des Definierens einer LUN zu erleichtern. Die allgemeine Funktionalität wird über Maus-Pop-Up, Tool-Bar-Buttons und Handlungsmenüs zur Verfügung gestellt, um auf einen existierenden Host zuzugreifen oder einen neuen Host zu definieren unter Verwendung beispielsweise eines "Füge einen neuen Host hinzu"-Buttons, eines "Verändere einen Host"-Buttons oder eines "Lösche einen Host"-Buttons.

Die Anwender-Schnittstelle besteht aus Menüs und einer Tabelle oder einem anderen grafischen Konstrukt zur Anzeige der Host-Information. Wenn der Anwender die Host-Verwaltungsoberfläche betritt, ist die Tabelle gefüllt mit allen existierenden Hosts. Der Anwender kann eine Reihe in der Tabelle auswählen. Jede Reihe enthält Information über einen Host. Der Anwender kann dann das Modifizieren oder Löschen des Hosts auswählen. Wenn das Modifizieren ausgewählt wird, erscheint eine Dialogbox, die dem Anwender ermöglicht, den Hostnamen und/oder die Beschreibung zu ändern. Der Anwender wird dann den OK- oder Abbruch-Button drücken. Wenn OK gedrückt wird, werden die Veränderungen in der Tabelle erscheinen und an den Server gesendet werden. Wenn Löschen ausgewählt wird, wird eine Dialogbox erscheinen mit einem Label, der den zu löschenden Host anzeigt und Buttons für OK oder Abbruch. Wenn OK gedrückt wird, wird die Hostzeile aus der Tabelle gelöscht und das Löschen wird beim Server durchgeführt. Wenn Hinzufügen ausgewählt wird, erscheint eine Dialogbox, die den Anwender in die Lage versetzt, alle Information über einen Host hinzuzufügen. Wenn OK ausgewählt wird, wird eine neue Reihe zu der Tabelle für diesen neuen Host hinzugefügt, und ein Hinzufügen wird beim Server ausgeführt. Das Klicken auf die Spaltenbezeichnung wird die Spalten sortieren.

Speicherabbildung

Der Anwender kann eine Speicherverwaltungsroutine starten, die ein Bild zeigt, das ein Fenster enthält zur Anzeige einer Darstellung eines hierarchischen Baums zum Anzeigen der Speicherelemente.

Speicherelemente werden definiert unter der Verwendung einer Baumstruktur (beispielsweise Spiegel zu Streifen zu Laufwerken). Dies ermöglicht dem Anwender, seinen Speicher in einer organisierten Weise aufzubauen, die konsistent ist mit ihrer Vorstellung über Speicher.

Repräsentative Typen von Speicherelementen umfassen die folgenden:

- Spiegel
- Streifen
- externe LUN
- internes Laufwerk
- SSD
- Speichersammlung
- Speicherpartition.

Durch das Aufbauen dieser Elemente in einem Baum (beispielsweise unter der Verwendung einer Microsoft Explorer-ähnlichen Baumanzeige) wird der Anwender in der Lage sein, Speicher zur Verwendung in virtuellen Verbindungen vorzukonfigurieren. Jedes Element kann partitioniert werden und diese Partitionen können auf verschiedene Arten verwendet werden. Beispielsweise kann ein Satz von Streifen partitioniert werden, wobei eine Partition als ein LUN exportiert wird und die andere als ein Mitglied in einem Spiegel verwendet wird (der daraufhin selbst partitioniert werden könnte).

Wenn ein Speicherelement partitioniert worden ist, werden die Partitionen in einer Speichersammlung gehalten, die das Kind des partitionierten Elementes ist. Für Elemente, die nicht partitioniert sind, wird diese Partitionssammlung nicht existieren. Jede Partition wird identifiziert durch den Typ von Speicher, den sie partitioniert, B beispielsweise eine Spiegelpartition, eine Laufwerkspartition, etc. Die Partitionen eines gegebenen Speicherelementes können nicht in eine einzelne Partition verschmolzen werden, außer alle Partitionen dieses Elementes stehen zur Verfügung (d. h. das gesamte Speicherelement ist unbenutzt). Zu diesem Zweck wird der Anwender ein partitioniertes Speicherelement auswählen, das nur nichtgenutzte Partitionen hat und den "Unpartition"-Button drücken.

Falls zugewiesene Reserven vorhanden sind, werden sie ebenfalls in einer Speichersammlung gehalten, die ein Kind des Elementes ist, dem diese Reservenzugeordnet sind.

Somit kann jedes Speicherelement potentiell die folgenden Kinder haben, eine Partitionssammlung, eine Reservesammlung und die tatsächlichen Speicherelemente, die das Elternelement umfassen.

Der Speichermanager ist in gewissem Sinne ein Blick in eine Speicher-Roll-Call-Tabelle, die allen verbundenen Speicher auf einem Server auflistet. Jedes verfügbare Speicherelement wird als der Kopf eines Speicherbaums gesehen. Beispielsweise wird ein Spiegel als verfügbar gezeigt werden, die Streifen und die Laufwerke, die die Zweige dieses Spiegels bilden, sind jedoch nicht verfügbar, da sie zu dem Spiegel gehören. Damit sie an anderer Stelle wiederverwendet werden, müßten sie von diesem Spiegel entfernt werden (und daher von dem Speicherbaum, der sich von diesem Spiegel aus erstreckt). In einem Ausführungsbeispiel wird dies getan über Drag and Drop in einer ähnlichen Weise, wie Dateien von einem Verzeichnis zu einem anderen im Windows NT-Dateiexplorer-Programm verschoben werden.

Der Baum des gesamten Speichers (verwendet und nicht verwendet) ist auf der linken Hälfte der Anzeige in diesem Beispiel gezeigt, wobei jedes Speicherelement ein Icon hat, das den Typ und irgendeinen Identifizierungsnamen oder eine ID darstellt.

Unterhalb des Baumes auf der rechten Seite des Fensters oder einem anderen geeigneten Platz wird die Liste des verfügbaren (nicht benutzten) Speichers gezeigt. Dies ist eine Liste des gesamten Speichers, der nicht durch ein anderes Speicherelement oder eine virtuelle Verbindung verwendet wird. Es wird erwartet, daß der meiste Speicher, der gegenwärtig nicht explizit verwendet wird, in einen generellen Reserve-Pool getan wird. Diese Liste des verfügbaren, nicht verwendeten Speichers, soll zumeist als eine Hilfe verwendet werden, damit der Anwender leicht nicht genutzte Speicherelemente findet zum Aufbau von neuen Speicherbäumen. Wenn beispielsweise eine Festkörperspeichergerät (SSD)-Partition gespiegelt wird durch eine Streifengruppe (RAID 0), werden die Partition und die Streifengruppe beide in der Verfügbarkeitsliste sichtbar sein, solange bis sie in den Spiegel eingefügt werden. Sobald der Spiegel aus den zwei Mitgliedern erzeugt ist, wird er in der Verfügbarkeitsliste zu sehen sein, solange bis er in eine virtuelle Verbindung eingefügt wird.

Auf der rechten Seite werden die Information und die Parameter, die einem beliebigen Element in dem Baum, das der Anwender durch einen Mausklick auswählt, zugeordnet sind, angezeigt werden. Wenn ein Speicherelement, das in der Verfügbarkeitsliste sichtbar ist, ausgewählt wird, wird es ausgewählt in sowohl der Verfügbarkeitsliste als auch dem Speicherbaum.

Funktionen zum Hinzufügen und Löschen werden bereitgestellt, um Einträge zu erzeugen oder zu entfernen, ebenso wie eine Modifizierungsfunktion, so daß unter der Verwendung der Werkzeuge, die von der Anwenderschnittstelle bereitgestellt werden, der Anwender Dinge wie "Eigentümer" oder "zuletzt gewartet" oder "Beschreibung", etc. in Feldern für Speicherelemente in dem Baum verändern kann. Der Anwender wird spezifizieren, was hinzugefügt wird (Spiegel, Streifen, Laufwerk, etc.) und eine geeignete Gruppe von Steuerungen wird ihnen gegeben.

Für ein internes Laufwerk und eine externe LUN wird der Anwender Dinge spezifizieren wie den Namen, die Größe, vielleicht den Hersteller. Das Spezifizieren eines inneren Laufwerks ist in gewissem Sinne ein Spezialfall, da ein Laufwerk ein Stück Hardware ist und daher automatisch detektiert würde. Der einzige Zeitpunkt, an dem Anwender ein Laufwerk hinzufügen würden, wäre, wenn sie einen Statthalter für irgendeine Hardware einfügen würden, die sie später hinzufügten. Dies kann ebenfalls für SSD-Boards durchgeführt werden.

Für RAID-Felder wird folgendes geschehen. Der Anwender wird spezifizieren, daß er ein Feld eines gegebenen

RAID-Niveaus erzeugen will (Spiegel oder Streifen anfangs) und wird dann in der Lage sein, die Speicherelemente zu spezifizieren, die die Mitglieder dieses Feldes sein werden. Diese Spezifizierung wird wahrscheinlich durch das Auswählen von Einträgen in einer Liste von verfügbaren Speicherelementen durchgeführt und die Feldkapazität wird durch die Kapazität seiner Mitglieder bestimmt werden. Die Speicherelemente, die als Mitglieder des Feldes benutzt werden, werden daraufhin als nicht verfügbar markiert (da sie Teil des Feldes sind) und das Feld selbst wird zu der Liste von verfügbarem Speicher hinzugefügt. Jedes RAID-Feld kann ferner bestimmte Reserven haben, die diesem Feld für den Fall zugewiesen werden, daß eines der Mitglieder ausfällt.

Speicherelemente können ferner partitioniert werden – dies geschieht durch das Auswählen des zu partitionierenden Elementes und durch das Spezifizieren, welche Stückgröße der Anwender haben möchte. Wenn das Element zuvor unpartitioniert war, wird dies dazu führen, daß zwei Partitionen erzeugt werden – die Partition, die der Anwender nachgefragt hat und eine weitere Partition, die den Rest (den nicht benutzten Bereich) des Speichers darstellt. Der nicht genutzte Bereich wird zusätzliche Partitionen ergeben, wenn sie erzeugt werden.

Die Detailanzeige für jedes Speicherelement wird so viel Information wie verfügbar ist anzeigen. Eines der Dinge, die in einem bevorzugten System gezeigt werden, ist, wie die Partitionen eines jeweiligen Speicherelementes aussehen (die Größe und die Position).

LUN-Abbildung

Unter der Verwendung eines Buttons auf der Anwenderschnittstelle wird eine Routine für eine LUN-Karte erzeugt. Die LUN (Logical Unit Number)-Karte ist im wesentlichen eine Liste der LUNs und ihrer zugeordneten Daten. Diese werden als eine Liste von Namen und Beschreibungen angezeigt. Die VC (virtuelle Verbindung), die einer gegebenen LUN zugeordnet ist, wird in dieser Anzeige gezeigt. Sie wird sichtbar gemacht, wenn der Anwender einen Eintrag aus der LUN-Karte auswählt und Details verlangt.

Die LUN-Karte wird die existierende Liste der LUNs zeigen mit Name, Beschreibung oder anderen Feldern. Die Felder umfassen:

- Name
- Beschreibung
- exportierter Status
- Host
- Speicherelement(e)

Die LUN-Karte ermöglicht:

- das Sortieren auf der Basis von verschiedenen Feldern.
- das Filtern, basierend auf Feldern. Dies ist nur nötig, wenn mehr als eine LUN zu einem Zeitpunkt bearbeitet wird (beispielsweise Einschalten/Ausschalten).
- Auswahl einer LUN zum Löschen oder Editieren/Ansehen.
- Definieren und Hinzufügen einer neuen LUN.
- Importieren von existierenden LUN(s), durchgeführt über "Learn Mode" beim Hardware-Start.
- Hinzufügen eines Mitgliedes und Starten eines Hot Copy-Spiegelprozesses auf einer LUN.
- Exportieren, Reexportieren einer LUN B. Dies wird im wesentlichen den Fluß Daten vom Host starten und stoppen.

Virtuelle Verbindungen sind (für den Anwender) definiert als ein Speicherbaum oder ein anderes grafisches Konstrukt, das mit einem Host verbunden ist, wie zum Beispiel die Dialogbox, die unter Verwendung eines Buttons gestartet wird. Die Dialogbox umfaßt ein Feld für den Eintrag eines LUN-Namens, ein Feld für den Eintrag einer Beschreibung und ein Feld für den Eintrag einer Ziel-ID und ein Feld für den Eintrag von Information über eine exportierte LUN. Pop Up-Menüs werden gestartet unter der Verwendung eines Host-Buttons für eine Liste von verfügbaren Hosts und ein Speicherbutton für eine Liste von verfügbaren Speicherelementen. Ein Zwischenspeicher-Auswahlbutton wird als eine Check Box implementiert.

Der Speicherbaum ist tatsächlich ein Baum von Speicherelementen (beispielsweise ein Spiegel, der irgendeine Anzahl von Streifengruppen umfaßt, die wiederum irgendeine Anzahl von Laufwerken umfassen). Der Host ist tatsächlich ein Server mit einer jeweiligen Initiator-ID, verbunden mit einem spezifischen Port auf einem NIC. Dies wird durch den Anwender über seine Auswahl eines vordefinierten Hosts und eines vordefinierten Speicherbaums definiert, der eine bestimmte Menge von verfügbarem Speicher repräsentiert.

Die Verwendung von Zwischenspeicher ist beschränkt auf "on" oder "off" unter der Verwendung einer Check Box. Alternative Systeme stellen Werkzeuge zur Spezifizierung von Zwischenspeichergröße und Zwischenspeicheralgorithmen bereit.

Die Verwendung von Zwischenspeicher kann im Betrieb ein- oder ausgeschaltet werden, ohne den Datenfluß entlang der virtuellen Verbindung zu unterbrechen. Die Standardeinstellung ist "on", wenn ein LUN erzeugt wird.

Ein Ausführungsbeispiel der LUN-Karte wird die Funktionalität haben, die notwendig ist zum Erzeugen von virtuellen Verbindungen. Sie wird aus einer mehrspaltigen Tabelle mit zwei Spalten bestehen; eine für Host und eine für Speicher. Die Erzeugung einer LUN wird sie automatisch exportieren und als verfügbare Funktionen "hinzufügen", "modifizieren" und "löschen" umfassen.

Die Anzeige der LUN-Karte ist ein Ort, an dem Hot Copy-Spiegel definiert werden, da dies üblicherweise mit einer existierenden LUN ausgeführt wird. Der Vorgang wird die folgenden Schritte umfassen: Auswählen der LUN, daraufhin Auswählen des Speicherbaums zum Hinzufügen zum bestehenden Speicherbaum über die Hinzufügung eines Spiegels

oder die Erweiterung eines existierenden Spiegels (beispielsweise Zweiwege zu Dreiwege).

Datenmigrationsunterstützung

Fig. 18 ist ein vereinfachtes Diagramm, das die drei Stufen des Datenflusses in einem Speichernetzwerk mit einem Speicherserver **10** zeigt, der mit dem ersten Speichergerät **11** über eine Kommunikationsverbindung **14** verbunden ist und mit einem zweiten Speichergerät **12** über eine Kommunikationsverbindung **15**. Das Zwischengerät **10** ist ebenfalls mit einem Client-Prozessor über eine Kommunikationsverbindung **13** verbunden, über die es eine Anforderung für Zugriff auf Daten einer logischen Adresse LUN A empfängt.

Der Speicherserver **10** umfaßt Speicher, wie zum Beispiel nichtflüchtigen Zwischenspeicher zur Verwendung als Puffer, Datentransfer-Ressourcen zum Transferieren von Datenzugriffsanforderungen, die auf der Verbindung **13** zu den Speichergeräten empfangen wird, auf die über die Verbindungen **14** und **15** zugegriffen werden kann.

Der Speicherserver umfaßt ferner eine Logikmaschine zur Verwaltung von Hot Copy-Vorgängen gemäß der vorliegenden Erfindung. Dieser Vorgang kann verstanden werden durch die Betrachtung der drei Stufen, die in **Fig.** 18 gezeigt sind.

In Stufe 1 bildet der Speicherserver **10** alle Datenzugriffsanforderungen, die die Datengruppe, die Gegenstand des Transfers ist, identifizieren und die auf der Schnittstelle zur Verbindung **13** empfangen werden, auf die Verbindung **14** zur Verbindung mit dem Gerät **11** ab, das die Datengruppe, die Gegenstand der Anforderung ist, speichert. Der Speicherserver empfängt ein Kontrollsignal, das einen Hot Copy-Vorgang startet und ein Zielgerät identifiziert, in diesem Beispiel das Gerät **12**. Dieser Schritt startet die Stufe 2, während der die Datengruppe als ein Hintergrundprozeß von dem ersten Gerät **11** über den Speicherserver **10** in das zweite Gerät **12** transferiert wird. Die Parameter werden auf dem Speicherserver **10** gehalten, um den Fortschritt des Transfers der Datengruppe anzuzeigen und zur Anzeige einer relativen Priorität des im Hintergrund laufenden Hot Copy-Vorgangs in bezug auf Datenzugriffsanforderungen von dem Client-Prozessor. Während des Hot Copy-Vorgangs werden Datenzugriffsanforderungen auf das erste Gerät **11** und das zweite Gerät **12** abgebildet, abhängig vom Fortschritt der Hot Copy und dem Typ der Anfrage. Ferner umfaßt der Speicherserver Ressourcen zum Zuordnen einer Priorität an den Hot Copy-Vorgang. Wenn die Priorität des Hot Copy-Vorgangs niedrig ist, erfährt der Client-Prozessor keine signifikante Verzögerung bei der Ausführung seiner Datenzugriffsanforderungen. Wenn die Priorität des Hot Copy-Vorgangs vergleichsweise hoch ist, kann der Client-Prozessor einige Verzögerung bei der Ausführung seiner Datenzugriffsanforderungen erfahren, aber der Hot Copy-Vorgang wird schneller abgeschlossen.

Nach dem Abschluß des Transfers der Datengruppe ist die Stufe 3 erreicht. In der Stufe 3 werden die Datenzugriffsanforderungen von dem Client-Prozessor, die an die Datengruppe adressiert sind, zu dem zweiten Gerät **12** über die Kommunikationsverbindung **15** geroutet. Das Speichergerät **11** kann von dem Netzwerk völlig entfernt werden oder für andere Zwecke verwendet werden.

In dem bevorzugten Ausführungsbeispiel umfaßt der Speicherserver **10** einen Speicherbereichsmanager, wie oben beschrieben.

Die Speichergeräte **11** und **12** können unabhängige Geräte umfassen oder logische Partitionen innerhalb einer einzelnen Speichereinheit. In diesem Fall führt der Hot Copy-Vorgang zu einer Migration der Daten von einer Adresse innerhalb der Speichereinheit zu einer anderen Adresse.

Die **Fig.** 19, 20, 21 und 22 erläutern verschiedene Aspekte einer Software-Implementierung eines Hot Copy-Vorgangs zur Ausführung in dem intelligenten Netzwerkserver, der oben beschrieben ist. In anderen Speicherservern, die für einen Hot Copy-Vorgang verwendet werden, werden Veränderungen in der Implementierung durchgeführt, zur Anpassung des jeweiligen Systems. Mehr Details der Komponenten einer virtuellen Verbindung, eines dauerhaften Tabellenspeichers und der Anwender-Interface-Strukturen werden mit Bezug auf die folgenden Figuren beschrieben.

Fig. 19 zeigt die grundlegenden Datenstrukturen, die in einem Hot Copy-Vorgang verwendet werden. Eine erste Struktur **300** wird eine UTILITY REQUEST STRUCTURE genannt. Eine zweite Struktur **351** wird eine UTILITY STRUCTURE genannt. Eine dritte Struktur **352** wird eine MEMBER STRUCTURE genannt. Die MEMBER STRUCTURE **352** dient zur Identifizierung einer jeweiligen Verbindung und ihres Status. Die MEMBER STRUCTURE **352** umfaßt Parameter wie zum Beispiel einen Identifizierer für eine virtuelle Verbindung (virtual circuit identifier, VD ID), eine logische Blockadresse (logic block address, LBA), die eine Blocknummer für einen Block von Daten, der gegenwärtig von der virtuellen Verbindung behandelt wird, enthält, eine Zählung der Anforderungen, die in einer Schlange stehen, für die virtuelle Verbindung und einen Statusparameter.

Die UTILITY STRUCTURE **351** enthält Parameter, die sich auf ein Dienstprogramm beziehen, das gegenwärtig ausgeführt wird, in diesem Fall ein Hot Copy-Dienstprogramm. Es speichert Parameter, wie zum Beispiel die Identifizierung einer Datenquellengruppe SOURCE ID, eine Identifizierung oder Identifizierungen eines Zielspeichergerätes oder -geräten für den Hot Copy-Vorgang DESTINATION ID(s), eine Schlange von Anforderungen, die in Verbindung mit dem Dienstprogramm ausgeführt werden sollen und Parameter, die den gegenwärtig behandelten Block und seine Größe anzeigen.

Die UTILITY REQUEST STRUCTURE **350** enthält eine Anforderung für den Hot Copy-Vorgang, inklusive einer Vielzahl von Parametern, die den Vorgang betreffen. Sie umfaßt beispielsweise einen Parameter STATUS, der den Status der Anforderung anzeigt, eine Vielzahl von Flags, die die Anforderung unterstützen, einen Pointer auf eine entsprechende UTILITY STRUCTURE, einen Parameter, der die Priorität der Anforderung in bezug auf Eingabe/Ausgabeanforderungen von den Client-Prozessoren anzeigt, eine Quellenmaske, die die Datengruppe in der Quelle identifiziert und eine Zielmaske, die einen Ort in einem Zielgerät identifiziert, auf den der Hot Copy-Vorgang die Datengruppe kopiert. In einem Ausführungsbeispiel gibt es eine Vielzahl von Zielmasken für eine einzelne Hot Copy-Anforderung. Wie ebenfalls in **Fig.** 19 gezeigt, wird eine logische Blockadresse (LBA) in der UTILITY REQUEST STRUCTURE gehalten, die ebenfalls in der MEMBER STRUCTURE gehalten wird, für einen aktuellen Block von Daten innerhalb der behandelten Datengruppe.

Um einen Hot Copy-Prozeß zu starten, wird eine Anwendereingabe aufgenommen, die die Erzeugung der UTILITY

REQUEST STRUCTURE verursacht. Der dauerhafte Tabellenspeicher in dem Speicherserver wird mit der Struktur, dem Status der Quell- und Zielgeräte aktualisiert und die virtuelle Verbindung, die der Datengruppe zugeordnet ist, wird überprüft, die Treiber vorbereitet, um den Hot Copy-Vorgang zu starten und die Statusparameter in verschiedenen Datenstrukturen werden gesetzt. Der Fortschritt des Hot Copy-Vorgangs wird in dem dauerhaften Tabellenspeicher für den Fall von Ausfällen gehalten. In diesem Fall kann der Hot Copy-Vorgang erneut gestartet werden unter der Verwendung von anderen Ressourcen innerhalb des Servers, unter der Verwendung der Kopie der Statusinformation und der Datenstrukturen, die in dem dauerhaften Tabellenspeicher gespeichert worden sind.

Die anderen Treiber in dem System, wie z. B. RAID-Monitore oder ähnliches, werden von dem Hot Copy-Vorgang in Kenntnis gesetzt.

Die Anforderung wird in die Schlange für die MEMBER STRUCTURE gestellt.

Sobald die Vorbereitung abgeschlossen ist, werden die Eingangs- und Ausgangsprozesse bei der Unterstützung des Hot Copy-Vorgangs gestartet. Die relative Priorität der Eingabe- und Ausgabeprozesse bei der Unterstützung des Hot Copy-Vorgangs bestimmen die Fortschrittsgeschwindigkeit des Hot Copy-Vorgangs für den Fall, daß ein Client-Prozessor Eingabe- und Ausgabeanforderungen für dieselbe Datengruppe ausführt. In der bevorzugten Ausführungsform werden Eingabe- und Ausgabeanforderungen von dem Client-Prozessor zuerst ausgeführt. Für den Fall, daß ein Blocktransfer bei der Unterstützung eines Hot Copy-Vorgangs gerade ausgeführt wird, wenn eine Eingabe- oder Ausgabeanforderung von einem Client-Prozessor empfangen wird, wird der Blocktransfer abgeschlossen als ein unteilbarer Vorgang und die Anforderung des Client-Prozessors wird daraufhin bedient. In alternativen Systemen können andere Techniken verwendet werden, um die Priorität der Vorgänge zu verwalten.

Der grundlegende Vorgang zur Ausführung einer Hot Copy ist in **Fig.** 20 dargestellt. Der Vorgang beginnt mit einer Hot Copy-Anforderung, die die Spitze der Schlange für die MEMBER STRUCTURE erreicht (Schritt **360**). Der Vorgang allokiert einen Puffer in dem Speicherserver zur Unterstützung des Blocktransfers (Schritt **361**). Eine Nachricht wird ausgegeben, um eine Kopie eines ersten Blocks in der Datengruppe in den Puffer zu verschieben (Schritt **362**). Ein aktueller Block wird zu dem Puffer verschoben gemäß der Prioritätseinstellung für den Hot Copy-Vorgang (Schritt **363**). Das Verschieben des Blocks wird erreicht unter der Verwendung von geeigneten Speicherverriegelungs-Transaktionen, um den Zugriff durch mehrere Prozesse innerhalb des Speicherservers zu steuern. Als nächstes wird eine Nachricht ausgegeben, eine Kopie des Blocks von dem Puffer zu dem Ziel oder den Zielen zu verschieben (Schritt **364**). Der Block wird zu dem Ziel oder den Zielen gemäß der Priorität für den Hot Copy-Vorgang verschoben (Schritt **365**). Sobald der Block verschoben ist, werden der dauerhafte Tabellenspeicher und die lokalen Datenstrukturen, die den Vorgang unterstützen, mit Statusinformation, die den Fortschritt der Hot Copy anzeigt, aktualisiert (Schritt **366**). Der Vorgang bestimmt, ob der letzte Block in der Datengruppe kopiert worden ist (Schritt **367**). Falls nicht, wird eine Nachricht ausgegeben, um eine Kopie des nächsten Blocks in den Puffer zu verschieben (Schritt **368**). Der Vorgang springt in einer Schleife zum Schritt **363**, um fortzufahren, Blocks der Datengruppe zu dem Ziel oder den Zielen zu verschieben. Wenn im Schritt **367** festgestellt wird, daß der letzte Block in der Datengruppe erfolgreich zu dem Ziel oder den Zielen verschoben worden ist, ist der Vorgang abgeschlossen (Schritt **369**).

Gemäß einem Ausführungsbeispiel der vorliegenden Erfindung ist es für einen Hot Copy-Vorgang, der mehrere Ziele mit sich bringt, möglich, daß ein Mitglied oder Mitglieder der Gruppe von Zielen, die gerade verwendet werden, während des Vorgangs ausfällt. In diesem Fall kann der Vorgang mit dem Ziel oder mit den Zielen, die weiterhin arbeiten, fortfahren durch das Aktualisieren der geeigneten Tabellen bei der Unterstützung des fortgesetzten Vorgangs.

Somit wird ein Hot Copy-Merkmal dazu verwendet, um eine Datengruppe von einem individuellen Mitglied, das noch nicht heruntergefahren ist, zu einem Ersatzlaufwerk zu kopieren. Die Datengruppe kann die gesamten Inhalte eines Speichergerätes umfassen oder irgendeinen Teil der Inhalte eines Speichergerätes. Die Hot Copy-Eigenschaft kann verwendet werden auf RAID-Feldern von irgendeinem Niveau mit geeigneter Status- und Parameterverwaltung.

Hot Copy-Parameter umfassen die Priorität des Vorgangs, das Quellgerät und einen Zielidentifizierer. Eine Hot Copy-Anforderung enthält eine Identifizierung des Quellmitglieds, eine Identifizierung des Zielmitglieds, die Copyblockgröße und die Copyfrequenz oder -Priorität. Hot Copies werden gemäß der Priorität, eine Blockgröße nach der anderen, durchgeführt. Die gegenwärtige Blockposition wird in Feldkonfigurationsdaten innerhalb der Datenstrukturen gehalten, wie oben erläutert. Der Hot Copy-Vorgang wird simultan mit normalen Eingangs- und Ausgangsprozessen durchgeführt. Beim Schreiben auf das Laufwerk, mit dem gerade eine Hot Copy durchgeführt wird, wird auf beide Laufwerke geschrieben. In diesem Fall ist das ursprüngliche Quellmitglied immer noch gültig, wenn die Hot Copy abgebrochen wird oder ausfällt. Wenn eine Hot Copy abgeschlossen ist, wird das ursprüngliche Quellmitglied von dem Feld entfernt und von Systemverwaltungsprogrammen als nicht verwendbar bezeichnet. In ähnlicher Weise wird in einem Ausführungsbeispiel das virtuelle Gerät, das die Datengruppe unterstützt, aktualisiert, um auf das neue Ziel zu zeigen.

Die **Fig.** 21 und 22 erläutern Vorgänge, die in dem Speicherserver ausgeführt werden, um Datenzugriffsanforderungen, die von den Client-Prozessoren ausgegeben werden, zu verwalten, während ein Hot Copy-Vorgang ausgeführt wird. Die Datenzugriffsanforderungen können irgendeinen einer Vielzahl von Typen haben, inklusive Leseanforderungen und Schreibanforderungen und Abwandlungen derselben. Andere Anforderungen umfassen Anforderungen zur Unterstützung der Verwaltung der Datenkanäle und ähnliches. In **Fig.** 21 ist ein Vorgang zur Behandlung einer Schreibanforderung erläutert.

Wenn eine Schreibanforderung die Spitze der Schlange erreicht, beginnt der Vorgang (Schritt **380**). Der Vorgang entscheidet, ob die Schreibanforderung einen Ort innerhalb der Datengruppe identifiziert, die Gegenstand eines aktuellen Hot Copy-Vorgangs ist (Schritt **381**). Wenn der Block innerhalb der Datengruppe ist, die gerade hot-kopiert wird, entscheidet der Prozeß, ob der Block, auf den die Schreibanforderung gerichtet ist, bereits zu dem Ziel kopiert worden ist (Schritt **382**). Wenn er kopiert worden ist, wird eine Nachricht ausgegeben, um sowohl auf die Speichergeräte, die die Datengruppe ursprünglich enthalten haben und auf das oder die Zielspeichergeräte zu schreiben (Schritt **383**). Als nächstes werden die Daten gemäß der Priorität für die Eingabe- und Ausgabeanforderung (Schritt **384**) verschoben und der Prozeß ist fertig (Schritt **385**).

Wenn in einem Schritt **381** die Anforderung nicht innerhalb der Datengruppe war, wird die Nachricht ausgegeben, das

Schreiben auf der Quelle der Datengruppe auszuführen (Schritt **386**). Der Ablauf des Vorgangs schreitet an diesem Punkt fort zum Schritt **384**. In ähnlicher Weise wird, wenn in einem Schritt **382** festgestellt wird, daß der Ort, der Gegenstand des Schreibens ist, nicht bereits kopiert worden ist, die Nachricht ausgegeben, auf das Quellgerät zu schreiben (Schritt **386**).

Fig. 22 erläutert die Behandlung einer Leseanforderung, die während einer Hot Copy erfolgt. Der Vorgang beginnt, wenn die Leseanforderung die Spitze der Schlange für das virtuelle Gerät erreicht (Schritt **390**). Der Vorgang entscheidet als erstes, ob das Lesen in die Datengruppe fällt, die Gegenstand der Hot Copy ist (Schritt **391**). Wenn das Lesen in die Datengruppe fällt, entscheidet der Prozeß, ob das Lesen in einen Block fällt, der bereits auf das Ziel oder die Ziele kopiert worden ist (Schritt **392**). Wenn festgestellt wird, daß der Lesevorgang innerhalb eines Blockes ist, der bereits auf das Ziel kopiert worden ist, wird eine Nachricht ausgegeben, die Daten vom neuen Ort zu lesen (Schritt **393**). In einem alternativen System kann das Lesen ausgeführt werden von dem Quellgerät oder von sowohl dem Quell- als auch den Zielgeräten in Abhängigkeit von der Zuverlässigkeit, Geschwindigkeit und anderen Faktoren, die die Verwaltung des Datenverkehrs innerhalb des Systems beeinflussen. Nach Schritt **393** werden die Daten an den Anfragenden zurückgegeben, gemäß der Priorität für die Datenzugriffsanforderungen des Client-Prozessors (Schritt **394**). Der Prozeß ist damit abgeschlossen (Schritt **395**).

Wenn in einem Schritt **391** festgestellt wird, daß die Leseanforderung nicht innerhalb der Datengruppe ist, die Gegenstand der Hot Copy ist, wird eine Nachricht ausgegeben, das Quellgerät zu lesen (Schritt **396**). Wenn in einem Schritt **392** festgestellt wird, daß die Leseanforderung einen Block adressiert, der noch nicht auf das Ziel kopiert worden ist, wird die Nachricht ausgegeben, die Daten von dem Quellgerät zu lesen (Schritt **396**). Nach Schritt **396** kehrt der Vorgang zurück zum Schritt **394**.

Für den Fall, daß eine Lese- oder Schreibanforderung auf Daten innerhalb eines speziellen Blocks auftritt, während der Block gerade durch den Speicherserverpuffer bewegt wird, werden Daten-Verschlußalgorithmen verwendet, um die Behandlung der Anforderungen zu verwalten. So wird beispielsweise, wenn ein logischer Block verschlossen wird bei der Unterstützung des Hot Copy-Vorgangs, während eine Lese- oder Schreibanforderung empfangen wird, dem Client-Prozessor mitgeteilt werden, daß die Lese- oder Schreibanforderung zurückgewiesen worden ist, da die Daten verschlossen werden. In alternativen Systemen, die eine höhere Priorität für den Client-Prozessor unterstützen, kann erlaubt werden, daß eine Lese- oder Schreibanforderung fortfährt, während der Block, der im Puffer gehalten wird, zur Unterstützung der Hot Copy gelöscht wird und der Status der Hot Copy wird zurückgesetzt, um anzuzeigen, daß der Block nicht bewegt worden ist. Eine Vielzahl von anderen Daten-Verschlußalgorithmen kann verwendet werden, wie für bestimmte Implementierungen benötigt.

Zielemulation

In den Konfigurationen, die in den **Fig. 1, 2** und 3 gezeigt sind, dient der Speicherserver als ein Zwischengerät zwischen Anwendern von Daten und Speichergeräten in dem Speicherbereich, die die Daten speichern. In dieser Umgebung wird zur Unterstützung von übernommenen Speichergeräten, d. h. Geräten, die vorhanden waren, bevor der Server als ein Zwischengerät eingefügt worden ist, der Speicher mit Ressourcen zum Emulieren des übernommenen Speichergerätes versehen. Auf diese Weise nimmt der Server virtuell die logische Adresse des übernommenen Gerätes gemäß dem zwischen dem Anwender und dem übernommenen Gerät verwendeten Speicherkanalprotokolls an, wenn der Server zwischen das übernommene Gerät und den Anwender der Daten eingefügt wird. Der Speicherserver dient dann dazu, auf alle Anforderungen gemäß diesem Protokoll zu antworten, die er empfängt und die an das übernommene Gerät adressiert sind. Ferner ruft der Speicherserver solche Konfigurationsinformation wie benötigt von dem übernommenen Gerät ab und speichert die Information im lokalen Speicher, so daß Status und Konfigurationsinformation, die der Anwender konfiguriert hat, um sie in dem übernommenen Gerät zu erwarten, bereitgestellt wird unter der Verwendung von lokalen Ressourcen auf dem Server. Dies spart Kommunikation zwischen dem Server und einem übernommenen Gerät und ermöglicht, daß der Server die Handlung des übernommenen Gerätes nachmacht gemäß dem Speicherkanalprotokoll, so daß die Rekonfiguration des Anwenders entweder nicht nötig ist oder stark vereinfacht wird nach dem Hinzufügen des Servers zum Speichernetzwerk.

Zusammenfassung

Speicherbereich-Netzwerke (SAN) sind eine neue speicherzentrierte Computerarchitektur. Zu großen Teilen veranlaßt durch die Verfügbarkeit von Faserkanalbasierten Speichersubsystemen und Netzwerkkomponenten versprechen SANs Datenzugriff und Bewegung mit hohen Geschwindigkeiten, flexiblere physikalische Konfiguration, verbesserte Ausnützung der Speicherkapazität, zentralisierte Speicherverwaltung, Online-Verwendung und Rekonfiguration der Speicherressourcen und Unterstützung für heterogene Umgebungen.

In dem älteren Modell einer "direkten Speicherzuordnung" hatten Speicherressourcen einen direkten Hochgeschwindigkeitspfad nur zu einem einzigen Server. Alle anderen Server hatte nur indirekt über ein LAN Zugriff mit wesentlich langsamerer Geschwindigkeit auf diese Speicherressource. Speicherbereich-Netzwerke verändern dies, indem sie direkte Hochgeschwindigkeits-Zugriffspfade (über den Faserkanal) von jedem Server zu jeder Speicherressource in einer "netzwerkartigen" Topologie bereitstellen. Die Einführung einer Netzwerkarchitektur verbessert ferner signifikant die Flexibilität der Speicherkonfiguration, das Entkoppeln von Speicherressourcen von einem jeweiligen Server und die Möglichkeit, verwaltet oder konfiguriert zu werden mit minimalem Einfluß auf die serverseitigen Ressourcen.

Während SANs die richtige Topologie bereitstellen, um die Anforderungen an die Flexibilität und den Datenzugriff in heutigen Umgebungen zu erfüllen, wird die SAN-Topologie selbst Geschäftsprobleme nicht in adäquater Weise gerecht. Nur physikalische Verbindungen zwischen Servern und Speicherressourcen über SAN-Gerüstkomponenten wie Schalter, Hubs oder Router bereitzustellen, ist nicht ausreichend, um das Versprechen des SANs voll zu erfüllen; jedoch stellt das SAN-Gerüst nicht die Hardware-Infrastruktur bereit zur Aufnahme der benötigten sicheren zentralisierten Speicher-

verwaltungsfähigkeit. Diese zwei Entwicklungen können, wenn sie gemeinsam verwendet werden, die Flexibilität und den allgegenwärtigen Zugriff auf essentielle Daten bereitstellen, die benötigt werden, um Geschäftsziele in der neuen Umgebung zu erfüllen.

Die Verwaltungsfähigkeit, die an der Spitze der SAN-Hardware-Infrastruktur benötigt wird, ist Speicherbereichsverwaltung. Um die optimale Speicherflexibilität und Zugriff mit hoher Leistung zu erreichen, ist die Speicherbereichsverwaltung am effizientesten innerhalb des SAN selbst angeordnet, anstatt in entweder den Servern oder den Speichergeräten. Server-basierte und Speicherressourcenbasierte Ansätze sind suboptimal, da sie nicht in adäquater Weise die Heterogenität sowohl auf der Server- als auch auf der Speicherseite unterstützen.

Speicherbereichsverwaltung ist eine zentralisierte und sichere Verwaltungsfähigkeit, die an der Spitze der existierenden SAN-Hardware-Infrastruktur angeordnet ist, um hohe Leistung, hohe Verfügbarkeit und fortgeschrittene Speicherverwaltungsfunktionalität für heterogene Umgebungen bereitzustellen. Der Zweck der Speicherbereichsverwaltung besteht darin, den Kern eines robusten SAN-Gerüstes zu bilden, das übernommene und neue Ausrüstung integrieren kann, SAN- und Speicherverwaltungsaufgaben von den Servern und Speicherressourcen auslagern kann und SAN-basierte Anwendungen aufnehmen kann, die über alle SAN-Komponenten verteilt sind. Ein SAN kann aufgebaut werden ohne die Verwendung von Speicherbereichsverwaltung, die Erzeugung und die Verwaltung einer optimierten heterogenen SAN-Umgebung benötigte jedoch diese entscheidende Verwaltungsfähigkeit.

Die Grundlagen von Speicherbereichsverwaltung umfassen:

- Heterogene Interoperabilität;
- sichere zentralisierte Verwaltung;
- Skalierbarkeit und hohe Leistungsfähigkeit;
- professionelle Zuverlässigkeit, Verfügbarkeit und Wartungsfähigkeit;
- eine intelligente zweckgerichtete Plattform.

Die Methode der Speicherbereichsverwaltung wird Kunden in die Lage versetzen, die vollen Fähigkeiten von SANs zu realisieren, um Geschäftsproblemen gerecht zu werden.

Bei all den Server- und Speicherzusammenstellungen ebenso wie den im heutigen Geschäftsklima üblichen Fusionen und Unternehmenskäufen ist Heterogenität eine Tatsache in einer Unternehmensumgebung. Eine Gruppe von Produkten, die SAN-Funktionalität für eine Produktlinie eines einzelnen Herstellers bereitstellt, ist nicht ausreichend, damit Kunden die vollen Fähigkeiten von SANs erreichen. Kunden benötigen eine Fähigkeit, die Investition in ältere Ausrüstung zu erhalten, selbst wenn sie neue Server und Speicherprodukte hinzufügen und nutzen. Daher muß ein Speicherbereichsmanager zumindest Faserkanal- und SCSI-Verbindungen unterstützen. Da sich der Speicherbereichsmanager mit der Zeit weiterentwickeln muß, um neue Technologien in dem Maße wie sie eingeführt werden, aufzunehmen, ist die Plattform in der Lage, einen wohldefinierten Wachstumspfad für extensivere Multiprotokollverbindungen mit der Zeit bereitzustellen.

SANs erzeugen einen großen virtualisierten Speicherpool, der zentral verwaltet werden kann, um Speicherverwaltungsaufgaben gegenüber der traditionellen Speicherarchitektur der "direkten Verbindung" zu minimieren, insbesondere in den Bereichen von Backup/Wiederherstellung und Totalausfall/Wiederherstellung. Da SANs effektiv einen physikalischen Zugriffspfad von allen Servern zu allen Speichern bereitstellen, jedoch nicht alle Server logisch auf alle Speicher zugreifen können sollten, muß Sicherheit auf eine robuste Weise angegangen werden. SAN-Gerüstanbieter erreichen dies durch die logische Definierung von "Zonen", wobei jeder Server nur in der Lage ist, auf Daten zuzugreifen, die als innerhalb seiner Zone definiert sind. Offensichtlich ist die Fähigkeit zur Definition von sicheren Zonen oder Speicher"bereichen" ein Aspekt eines Speicherbereichsmanagers. Eine verbesserte Unterteilbarkeit von Bereichsdefinitionen, wie zum Beispiel die Definition von Untereinheiten innerhalb einer Zone auf dem LUN-Niveau anstelle des Port-Niveaus bietet signifikante zusätzliche Flexibilität bei der Verbesserung der Speicherausnutzung über die Zeit. Der Speicherbereichsmanager bietet eine vollständige Gruppe von zentralisierten Speicherverwaltungsfähigkeiten, die von einer einzigen Verwaltungsschnittstelle über alle verbundenen Server und Speicher unabhängig vom Hersteller verwendet werden kann. Von einem zentralen Ort kann ein Systemadministrator das Verschieben oder Spiegeln von Daten zwischen heterogenen Speicherressourcen steuern und dynamisch diese Fähigkeiten über verschiedene heterogene Speicherressourcen über die Zeit verteilen. Dies führt zu signifikanten Kostenersparungen und der Vereinfachung der Verwaltungskomplexität. Als eine skalierbare intelligente Plattform sitzt der Speicherbereichsmanager in dem perfekten zentralen Ort, um Speicherverwaltungsfunktionalität aufzunehmen, die über alle verbundenen Server und Speicherressourcen verteilt werden kann.

Bei den gegebenen Speicherwachstumsraten, verursacht durch das neue Geschäftsklima, kann eine spezifische SAN-Umgebung leicht während ihrer Lebensdauer um zwei Größenordnungen an Speicherkapazität wachsen. Als Spitze der zentralen Intelligenz in dem SAN ist ein Speicherbereichsmanager in der Lage, eine signifikante Menge an Wachstum zu verkraften, ohne eine belastungsbezogene Leistungsverschlechterung. Intelligenz sollte hinzugefügt werden in dem Maße wie die Konfiguration wächst, um eine glatte, kosteneffektive Skalierbarkeit über einen breiten Leistungsbereich sicherzustellen.

Eine Fähigkeit zur Zwischenspeicherung von signifikanten Datenmengen optimiert in der intelligenten Plattform die SAN-Konfiguration, um Leistungsverbesserungen in anwendungsspezifischen Umgebungen zu erreichen. Wenn beispielsweise "Hot Spots", wie zum Beispiel Dateisystem-Journale und Datenbankregister oder Protokolldateien in einem Hochgeschwindigkeitsspeicher in dem Speicherbereichsmanager selbst zwischengespeichert werden können, minimiert dies in signifikanter Weise die Nachrichtenpfadverzögerung im Vergleich zu mehr konventionellen SAN-Konfigurationen, die ohne einen Speicherbereichsmanager aufgebaut sind. Unter der Annahme einer ausreichenden Menge von Onboard-Speicher können ganze Datenbanken und Dateisysteme effektiv zwischengespeichert werden, um große Leistungsverbesserungen zu erreichen. Die Onboard-Speicherkapazität ist ferner wichtig zum Einspeichern von Daten während der Migration und anderer Aufgaben der Datenverschiebung.

Wie bereits erwähnt, ist einer der entscheidenden Gründe für den Übergang zu einem SAN, die allgemeine Datenverfügbarkeit zu verbessern. Wenn einzelne Ausfallpunkte als ein Ergebnis des Übergangs zu dieser neuen Speicherarchitektur eingefügt werden, werden viele ihrer möglichen Vorteile nicht realisiert. Aus diesem Grund müssen nicht nur die Daten selbst, sondern auch die Zugriffspfade zu diesen Daten zu jedem Zeitpunkt verfügbar sein. Die Minimierung von Ausfallzeit aufgrund von Ausfällen muß angegangen werden durch die Verwendung von relativen internen Komponenten und Fähigkeiten wie automatische I/O-Pfadausfallübernahme, logischem schnellen Austausch (hot sparing) und einsteckbaren, unmittelbar austauschbaren (hot swappable) Komponenten. Die Ausfallzeit muß ferner minimiert werden durch Online-Verwaltungsfähigkeiten, wie zum Beispiel die Online-Aktualisierung von Firmware, dynamische Hardware- und Software-Rekonfiguration und hochleistungsfähiger Datenverschiebung im Hintergrund. Um die höchsten Leistungsniveaus sicherzustellen, ist der bevorzugte Speicherbereichsmanager eine zweckgebaute Plattform, die speziell für speicherbezogene Aufgaben, die von ihm verlangt werden, optimiert ist. Diese Plattform unterstützt signifikante lokale Verarbeitungsleistung zur Durchführung eines großen Bereichs von Speicherverwaltungsaufgaben, unterstützt durch den lokalen Hochgeschwindigkeitsspeicher, der notwendig ist für die Datenbewegung und die Ausführung der Anwendung zur Speicherverwaltung.

Im Vergleich mit einer Mehrzweckplattform, die als ein intelligenter Speicherserver verwendet wird, bietet eine für diesen Zweck gebaute Plattform ein Realzeit-Betriebssystem für eine schnellere und besser bestimmte Antwortzeit, effizienteren I/O-Pfadcode zur Minimierung von Nachrichtsverzögerungen und einen Betriebssystem-Kernel, der optimiert ist als eine Datenverschiebungsmaschine anstelle einer Anwendungsmaschine.

Diese für diesen Zweck gebaute Plattform unterstützt Merkmale auf Kernel-Ebene, die in einem Mehrzweck-Betriebssystem nicht zur Verfügung stehen, wie zum Beispiel die zuverlässige deterministische Lieferung von Nachrichten. Die Merkmale der hohen Verfügbarkeit, wie zum Beispiel eine integrierte Pfadausfallübernahme, die Online-Verwaltung und die dynamische Rekonfigurierung werden durch das Kern-Betriebssystem unterstützt. Durch das Bereitstellen von Intelligenz an dem optimalen Ort zur Unterstützung der heterogenen SAN-Umgebungen bringt der Speicherbereichsmanager die folgenden Geschäftsvorteile für Endanwender:

- verbesserte Speicherressourcenzuweisung und -ausnutzung;
- die Flexibilität, um kosteneffizient dynamische Speicherumgebungen mit hohem Wachstum aufzunehmen;
- eine hohe Verfügbarkeit durch Online-Verwaltung und -Konfiguration;
- effizientere Verwaltung, um die gesamten $/GB-Kosten der Speicheradministration zu senken;
- eine Fähigkeit, um heterogene Server und Speicher in einer integrierten SAN-Umgebung zu verbinden;
- das Erhöhen des Wertes des JBOD-Speichers durch das Hinzufügen von Merkmalen der Speicherverwaltung und des Zwischenspeicherns, die dynamisch über alle Speicherressourcen verteilt werden können.

Eine robuste SAN-Hardware-Infrastruktur, die gemeinsam mit der Methode der Speicherbereichsverwaltung verwendet wird, stellt die Flexibilität zur Aufnahme einer sich schnell und nicht vorhersagbar ändernden Umgebung bereit und stellt gleichzeitig sicheren Hochgeschwindigkeitszugriff auf hochverfügbare Daten bereit. Das resultierende zentralisierte Speicherverwaltungsparadigma ist ein effizienterer billigerer Weg zur Verwaltung des Wachstums von Daten, die den Wettbewerbsvorteil für das Unternehmen begründen.

Die vorangegangene Beschreibung von zahlreichen Ausführungsbeispielen der Erfindung ist zur Erläuterung und Beschreibung dargelegt worden. Die Beschreibung ist nicht beabsichtigt, die Erfindung auf die exakt offenbarten Formen zu begrenzen. Viele Veränderungen und äquivalente Anordnungen werden für Fachleute erkennbar sein.

Patentansprüche

1. System zur Verwaltung von Speicherbereichen in einem Speichernetzwerk, wobei das Speichernetzwerk einen oder mehrere Clients und ein oder mehrere Speichersysteme umfaßt, wobei der eine oder die mehreren Clients entsprechende Speicherkanalprotokolle ausführen, die Information übertragen, die ausreichend ist zur Identifizierung eines Clients, der durch einen Speichervorgang bedient wird, aufweisend:
eine Vielzahl von Kommunikationsschnittstellen, die für eine Verbindung über Kommunikationsmedien zu entsprechenden anderen des einen oder der mehreren Clients oder des einen oder der mehreren Speichersysteme geeignet sind, und die gemäß der verschiedenen Kommunikationsprotokolle arbeiten;
eine Verarbeitungseinheit, die mit der Vielzahl von Kommunikationschnittstellen verbunden ist und Logik umfaßt zum Konfigurieren einer Gruppe von Speicherorten aus dem einen oder den mehreren Speichersystemen als ein Speicherbereich für eine Gruppe von zumindest einem Client aus dem einen oder den mehreren Clients, Logik zum Routen von Speichervorgängen innerhalb eines Speicherbereichs in Antwort auf den identifizierten Client;
Logik zum Übersetzen eines Speichervorgangs, der die Vielzahl der Kommunikationsschnittstellen durchläuft in und aus einem gemeinsamen Format; redundante Ressourcen, inklusive nichtflüchtigem Zwischenspeicher, um Speichervorgänge in dem gemeinsamen Format unter den Kommunikationsschnittstellen innerhalb des Speicherbereichs zu routen; und
eine Verwaltungsschnittstelle, die mit der Verarbeitungseinheit verbunden ist, zur Konfiguration des Speicherbereichs.

2. Das System nach Anspruch 1, wobei der eine oder die mehreren Clients entsprechende Speicherkanalprotokolle ausführen, die Information enthalten, die ausreichend ist zur Identifizierung eines logischen Speicherortes und Logik umfassen zum Routen von Speichervorgängen innerhalb eines Speicherbereichs in Antwort auf den logischen Speicherort.

3. System nach Anspruch 1, aufweisend Logik zur Verwaltung von Migration von Datengruppen von einem Speicherort zu einem anderen Speicherort innerhalb des Netzwerks.

4. System nach Anspruch 1, wobei die Verwaltungsschnittstelle Ressourcen zur Konfiguration einer Vielzahl von

Speicherbereichen mit dem Netzwerk umfaßt.

5. Verfahren zur Konfiguration und zur Verwaltung von Speicherressourcen in einem Speichernetzwerk, aufweisend:

Installieren eines Zwischensystems in dem Netzwerk zwischen Clients und Speicherressourcen in dem Netzwerk;

Zuweisen eines logischen Speicherbereichs zu Clients in dem Netzwerk unter der Verwendung von Logik in dem Zwischensystem;

Zuweisen von Speicherressourcen in dem Netzwerk zu logischen Speicherbereichen unter der Verwendung von Logik in dem Zwischensystem; und

Routen von Speichervorgängen durch das Zwischengerät gemäß den logischen Speicherbereichen, die den Clients zugeordnet sind und gemäß den Speicherressourcen, die den logischen Speicherbereichen zugeordnet sind.

6. Speicherserver, aufweisend:

eine Kommunikationsschnittstelle, die einen Kommunikationskanal für einen Speichervorgang unterstützt;

eine Logik zum Übersetzen eines Speichervorgangs, der über den Kanal für den Speichervorgang empfangen worden ist, in ein internes Format;

eine Logik zum Routen des Speichervorgangs in dem internen Format zu einer virtuellen Verbindung, wobei die virtuelle Verbindung Verbindungen zu entsprechenden Datenspeichern in Kommunikation mit dem Speicherserver verwaltet.

7. Speicherserver nach Anspruch 6, wobei die virtuelle Verbindung Logik zum Übersetzen des internen Formats in ein oder mehrere Kommunikationsprotokolle für einen oder mehrere entsprechende Datenspeicher umfaßt.

8. Speicherserver nach Anspruch 7, wobei die entsprechenden Kommunikationsprotokolle für entsprechende Datenquellen ein Protokoll umfassen, das mit einem standard-"intelligenten Eingabe/Ausgabe" ("intelligent input/output, I_2O")-Nachrichtenformat kompatibel ist.

9. Speicherserver nach Anspruch 6, wobei die Logik zum Routen von Speichervorgängen zu einer virtuellen Verbindung eine Tabelle umfaßt, wobei die Tabelle eine Vielzahl von Einträgen hat, wobei die Vielzahl der Einträge eine Übereinstimmung zwischen einem Adreßbereich, der in dem Speicherkommunikationskanal spezifiziert ist, und einer virtuellen Verbindung anzeigen.

10. Speicherserver nach Anspruch 6, wobei die Logik zum Routen von Speichervorgängen zu einem virtuellen Gerät eine Tabelle umfaßt, wobei die Tabelle eine Vielzahl von Einträgen hat, wobei die Vielzahl der Einträge eine Übereinstimmung zwischen einer virtuellen Verbindung und entsprechenden Datenquellen anzeigen.

11. Speicherserver nach Anspruch 6, aufweisend einen Zwischenspeicher, und wobei eine virtuelle Verbindung mit dem Zwischenspeicher kommuniziert.

12. Speicherserver nach Anspruch 6, wobei entsprechende Datenspeicher einen nichtflüchtigen Datenspeicher umfassen.

13. Speicherserver nach Anspruch 6, wobei entsprechende Datenspeicher ein Feld von Festplatten umfassen.

14. Speicherserver nach Anspruch 6, aufweisend eine Anwenderschnittstelle zur Unterstützung der Eingabe von Konfigurationsdaten.

15. Speicherserver nach Anspruch 14, wobei die Anwenderschnittstelle eine grafische Anwenderschnittstelle umfaßt.

16. Der Speicherserver nach Anspruch 14, wobei die Anwenderschnittstelle einen Touch Screen umfaßt, der mit dem Speicherserver verbunden ist.

17. Server für ein Speichernetzwerk mit zumindest einem Client-System, das Anforderungen für Speichervorgänge erzeugt, einem Client-Kommunikationskanal zu und von dem Client-System, einer Vielzahl von Speichergeräten und entsprechenden Kommunikationskanälen zu und von der Vielzahl von Speichergeräten, aufweisend:

einen Prozessor inklusive eines Bussystems;

eine Client-Schnittstelle zum Client-Kommunikationskanal, die mit dem Bussystem verbunden ist;

eine Vielzahl von Schnittstellen zu entsprechenden Kommunikationskanälen, die mit dem Bussystem verbunden sind;

einen nichtflüchtigen Zwischenspeicher, der mit dem Bussystem verbunden ist; und

Ressourcen, die von dem Prozessor gesteuert werden, um Anforderungen für Speichervorgänge auf der Serverschnittstelle zu empfangen, um die angeforderten Speichervorgänge an die Vielzahl von Speichergeräten zu leiten und um den nichtflüchtigen Zwischenspeicher zur Verwendung in den Speichervorgängen zu allokieren.

18. Server nach Anspruch 17, wobei die von dem Prozessor gesteuerten Ressourcen Prozesse umfassen zur Authentifizierung und Verifizierung von Zugangserlaubnissen für Speichervorgänge.

Hierzu 17 Seite(n) Zeichnungen

Nummer: DE 100 14 448 A1
Int. Cl.[7]: G 06 F 15/173
Offenlegungstag: 16. November 2000

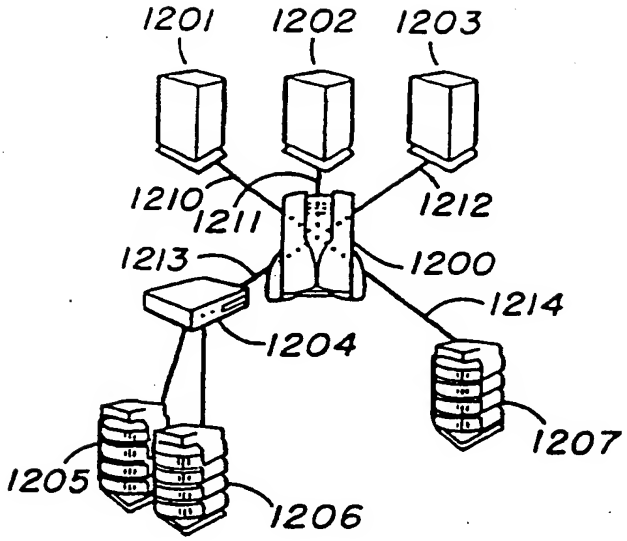


FIG. 1

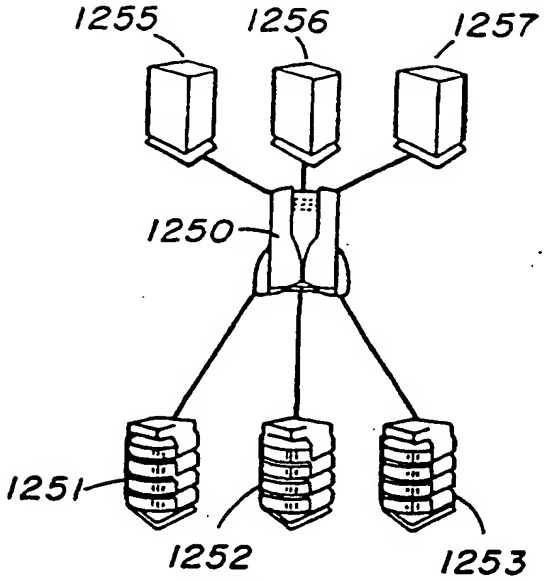


FIG. 2

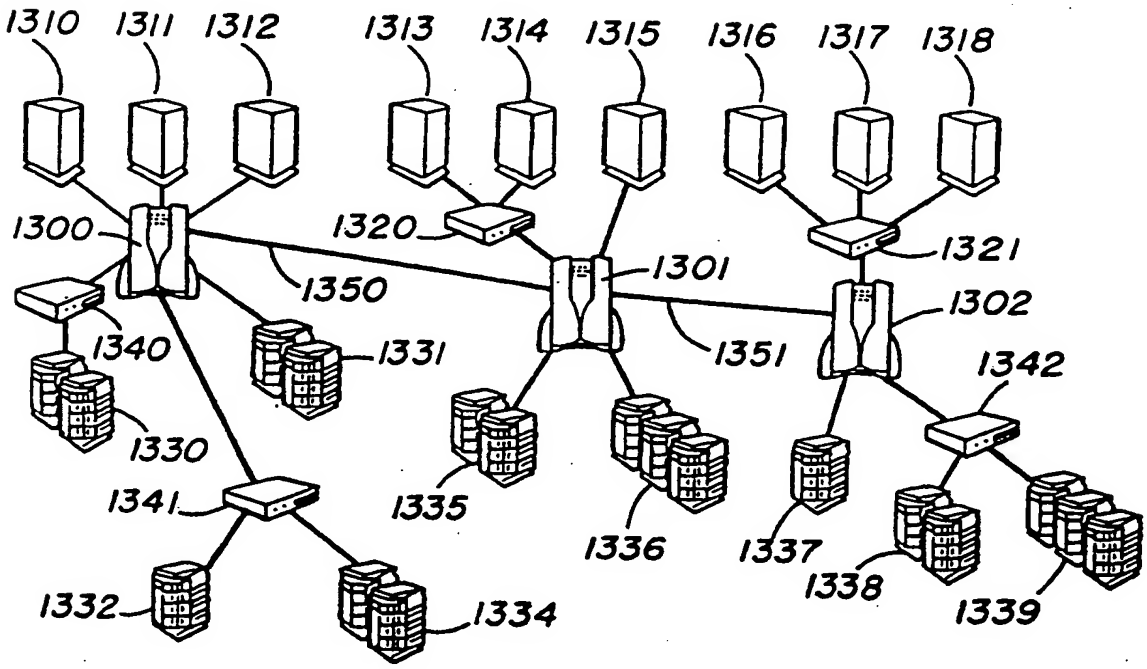
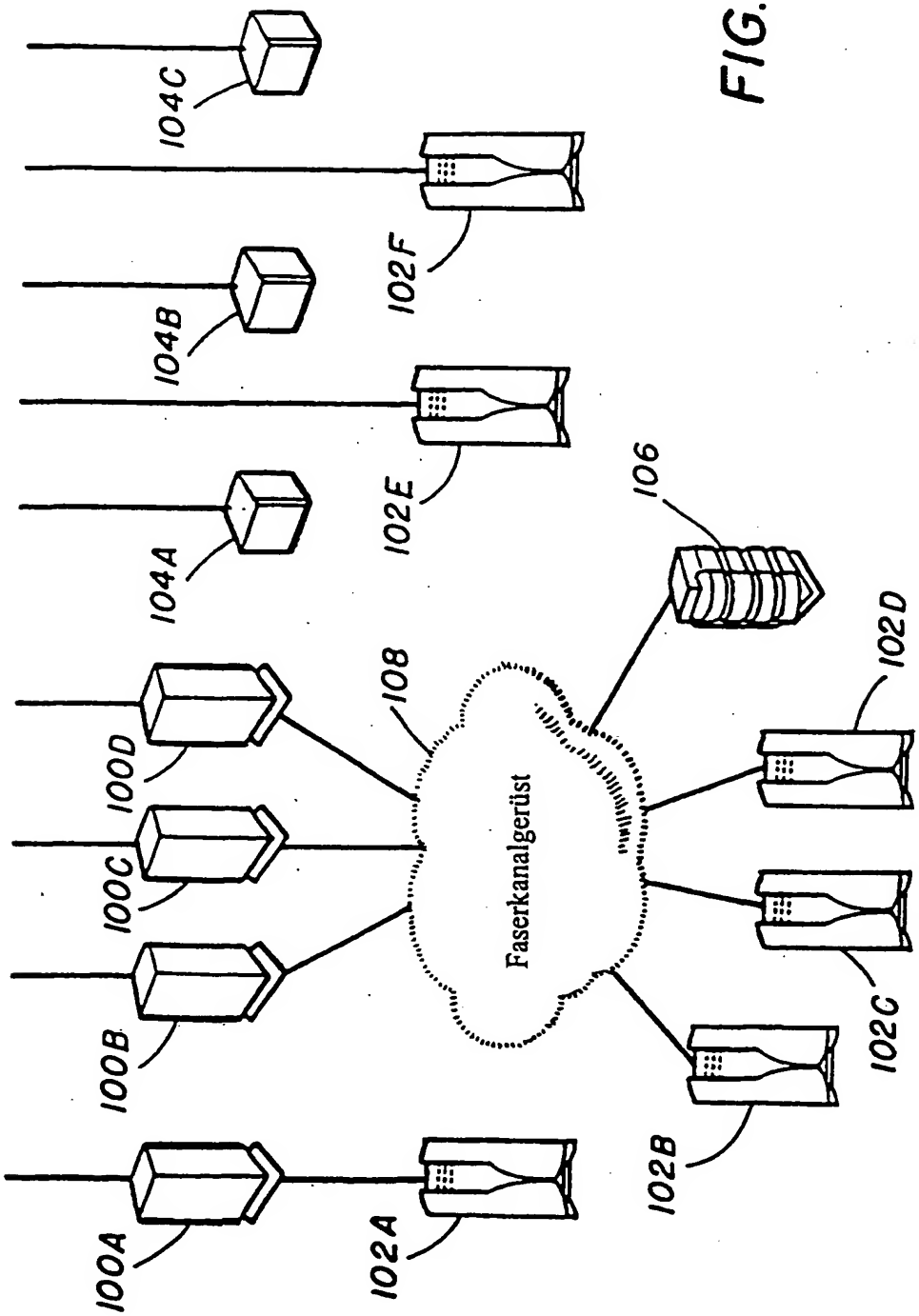


FIG. 3

Nummer: DE 100 14 448 A1
Int. Cl.[7]: G 06 F 15/173
Offenlegungstag: 16. November 2000

FIG. 1A



Nummer: DE 100 14 448 A1
Int. Cl.[7]: G 06 F 15/173
Offenlegungstag: 16. November 2000

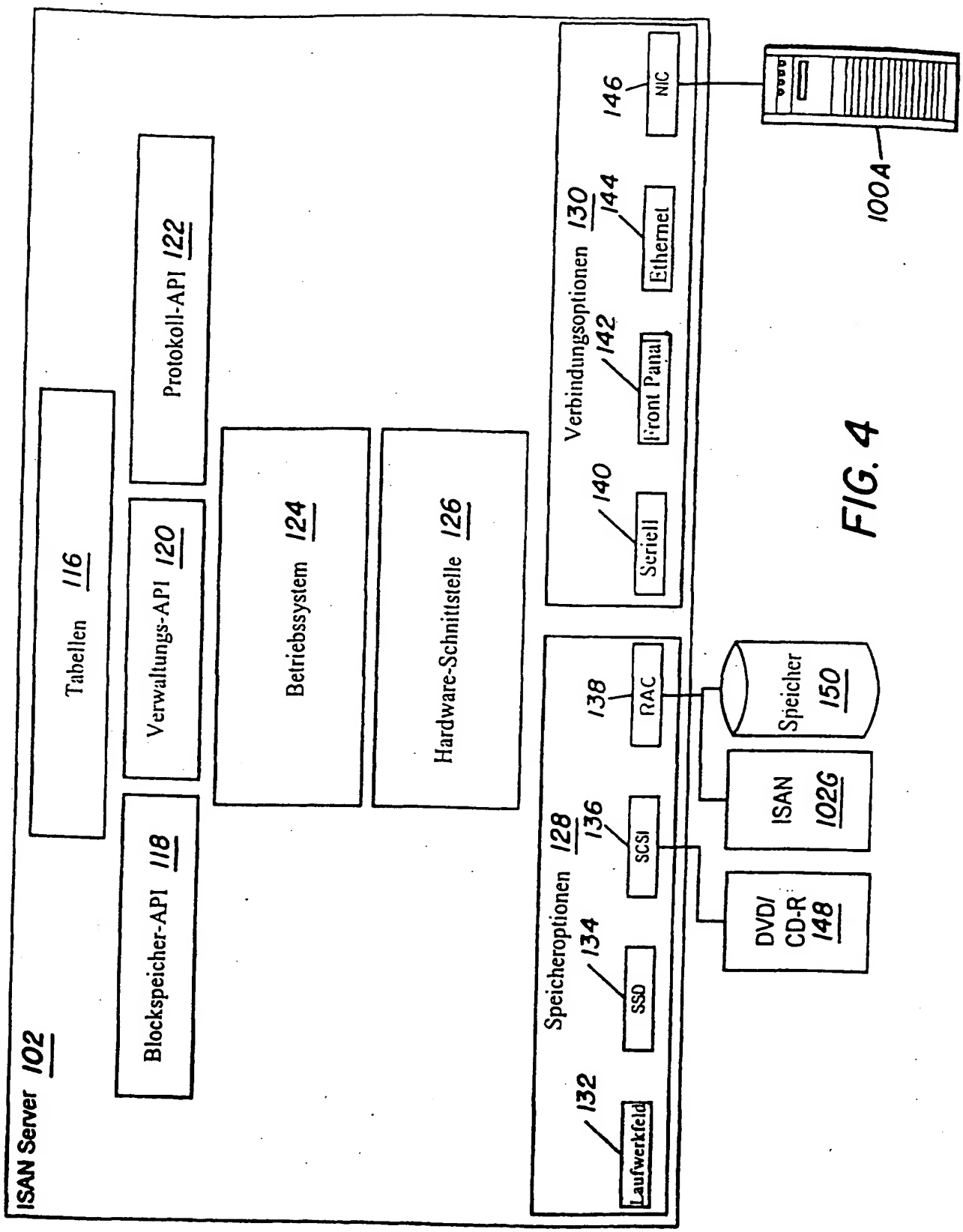
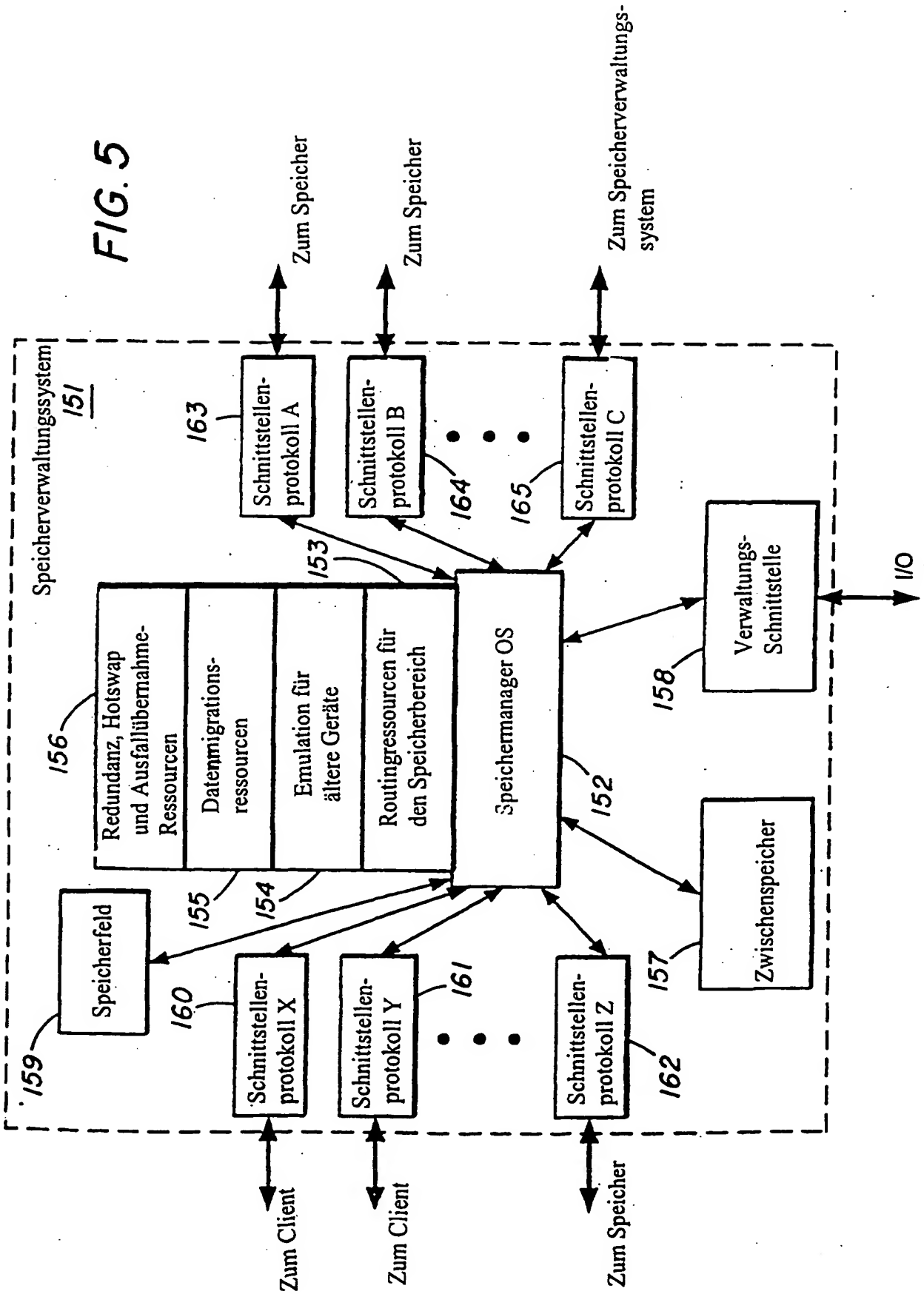


FIG. 4

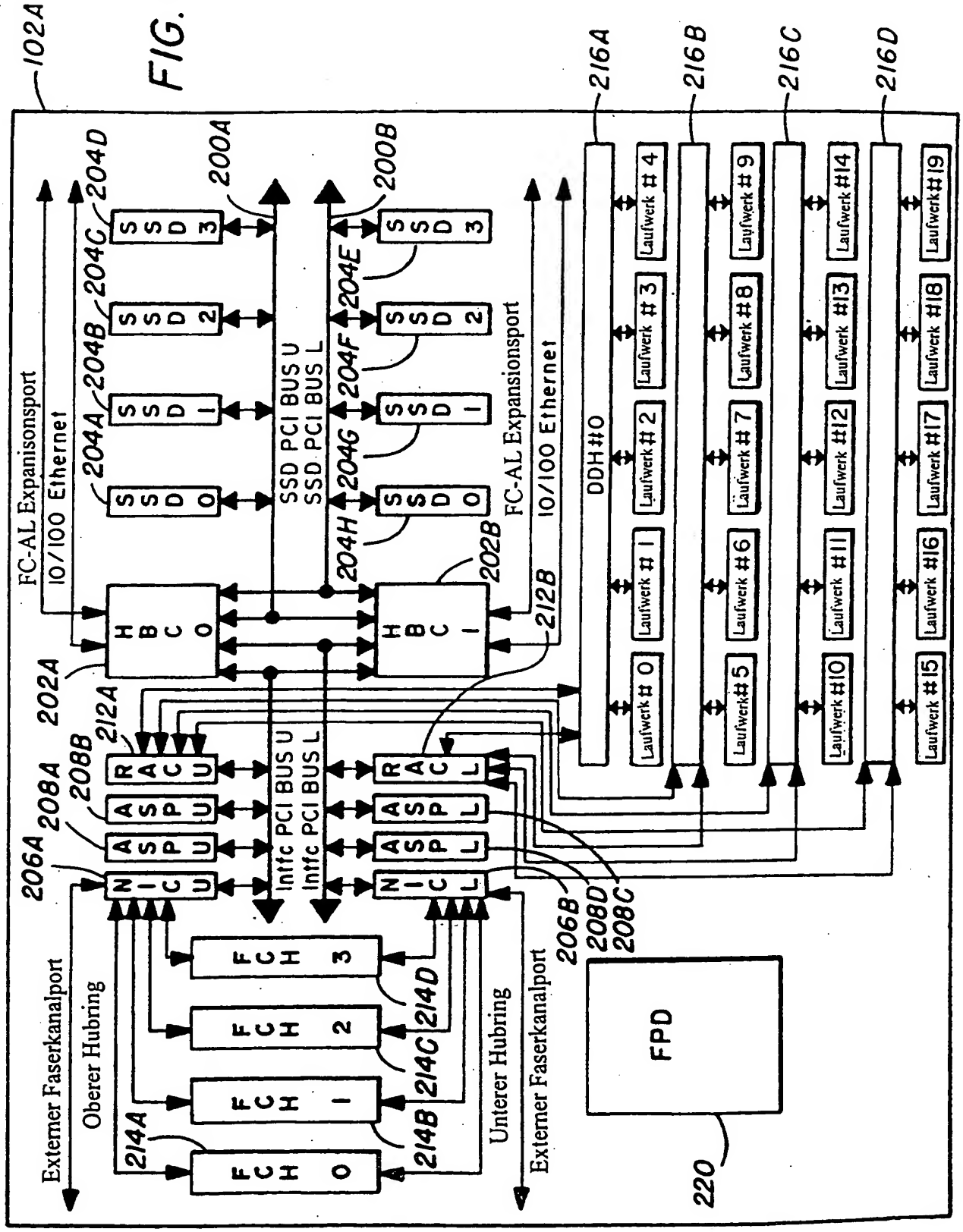
Nummer: DE 100 14 448 A1
Int. Cl.[7]: G 06 F 15/173
Offenlegungstag: 16. November 2000

FIG. 5



Nummer: DE 100 14 448 A1
Int. Cl.[7]: G 06 F 15/173
Offenlegungstag: 16. November 2000

FIG. 6



Nummer: DE 100 14 448 A1
Int. Cl.[7]: G 06 F 15/173
Offenlegungstag: 16. November 2000

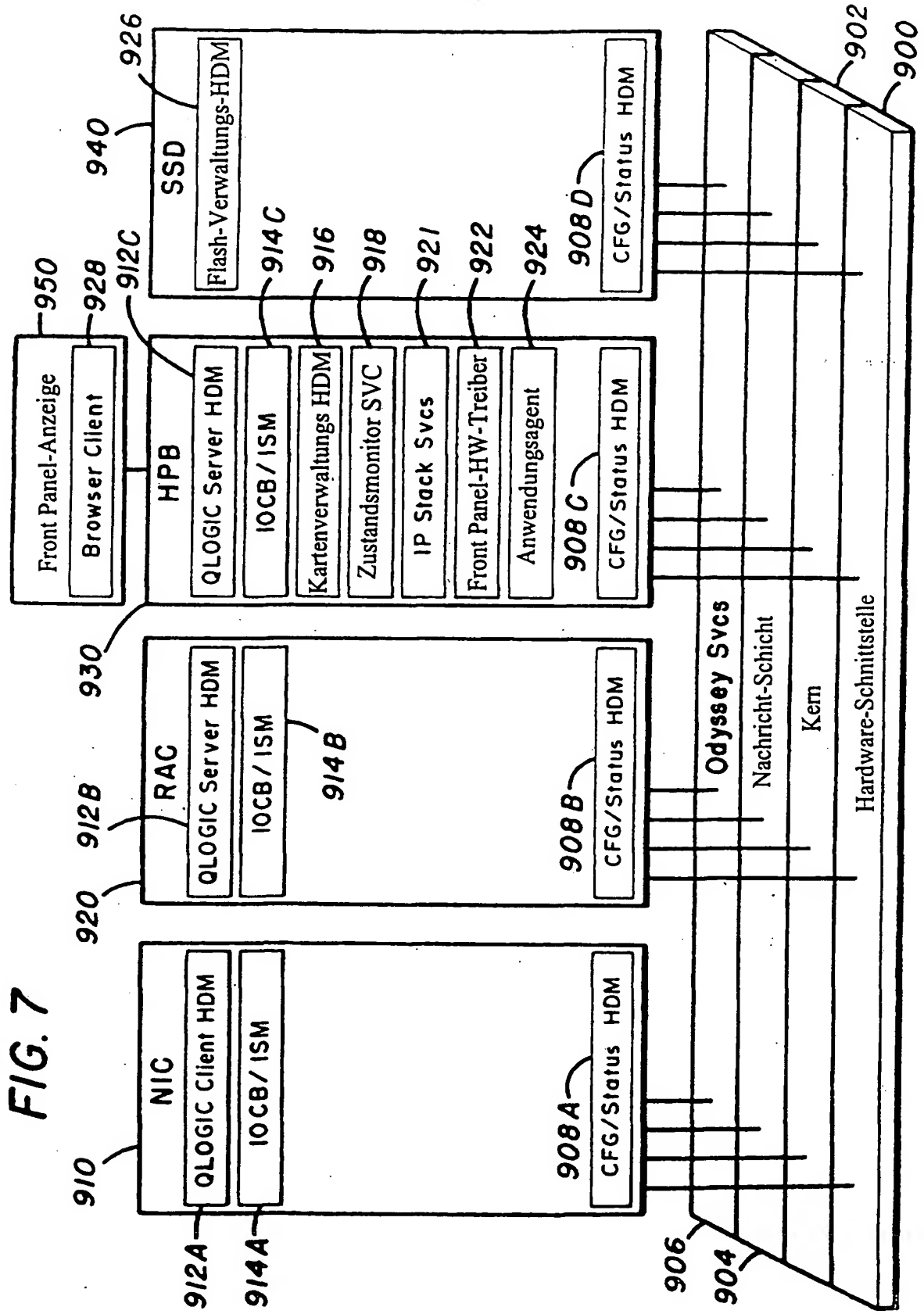


FIG. 7

Nummer: DE 100 14 448 A1
Int. Cl.[7]: G 06 F 15/173
Offenlegungstag: 16. November 2000

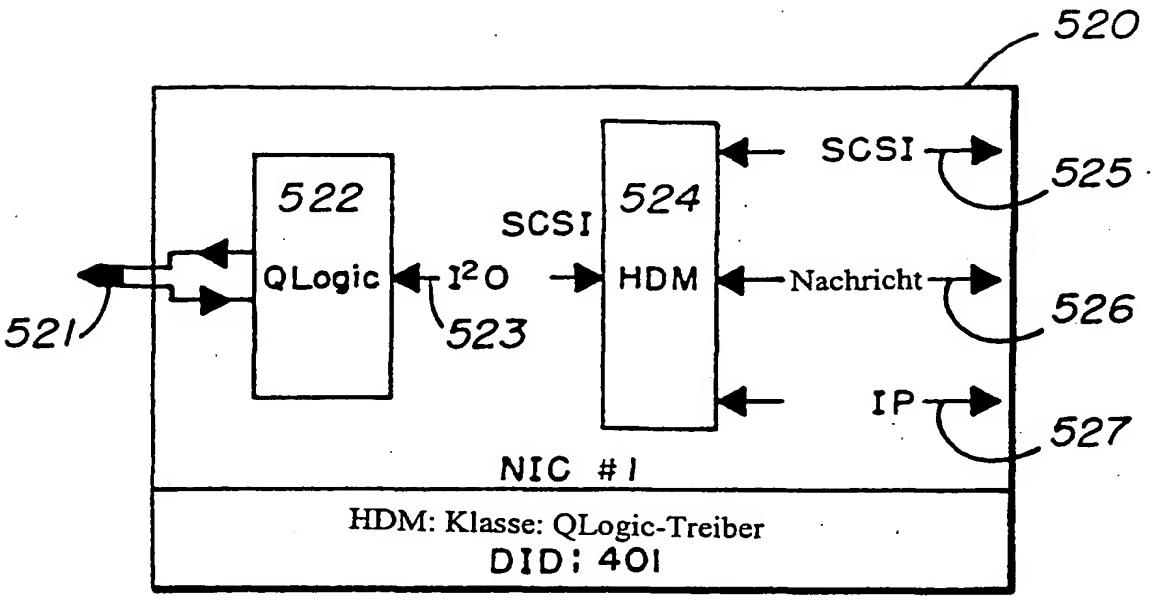


FIG. 8

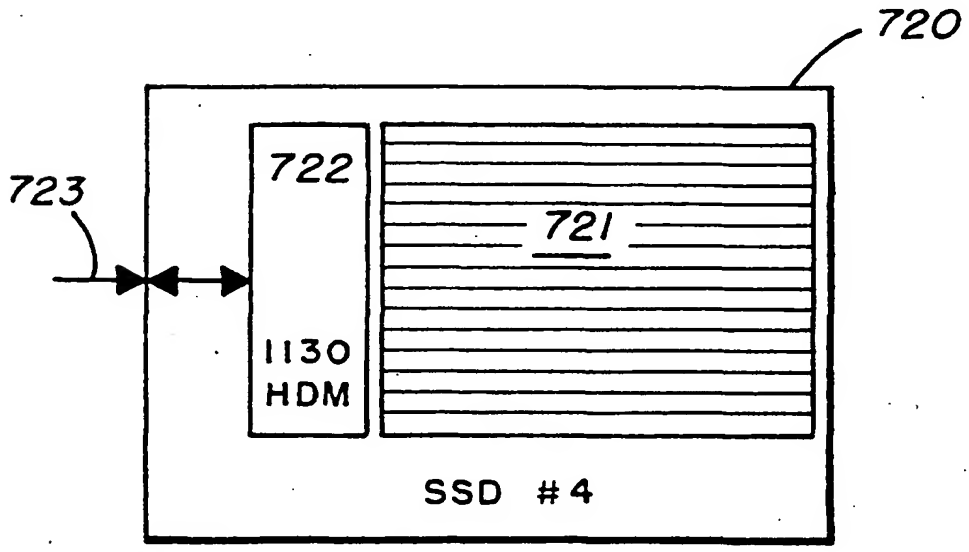
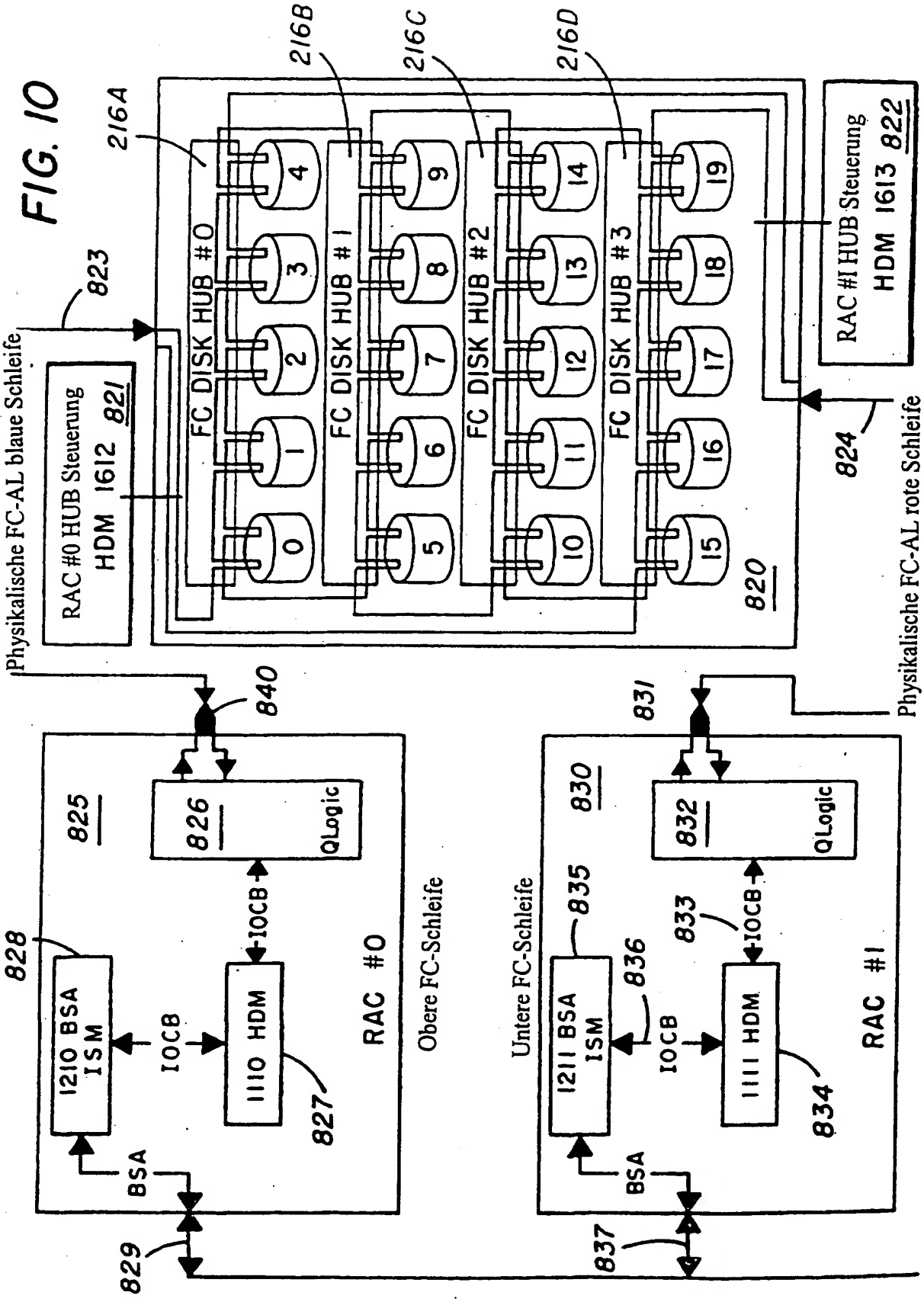


FIG. 9

Nummer: DE 100 14 448 A1
Int. Cl.[7]: G 06 F 15/173
Offenlegungstag: 16. November 2000

FIG. 10



Nummer: DE 100 14 448 A1
Int. Cl.[7]: G 06 F 15/173
Offenlegungstag: 16. November 2000

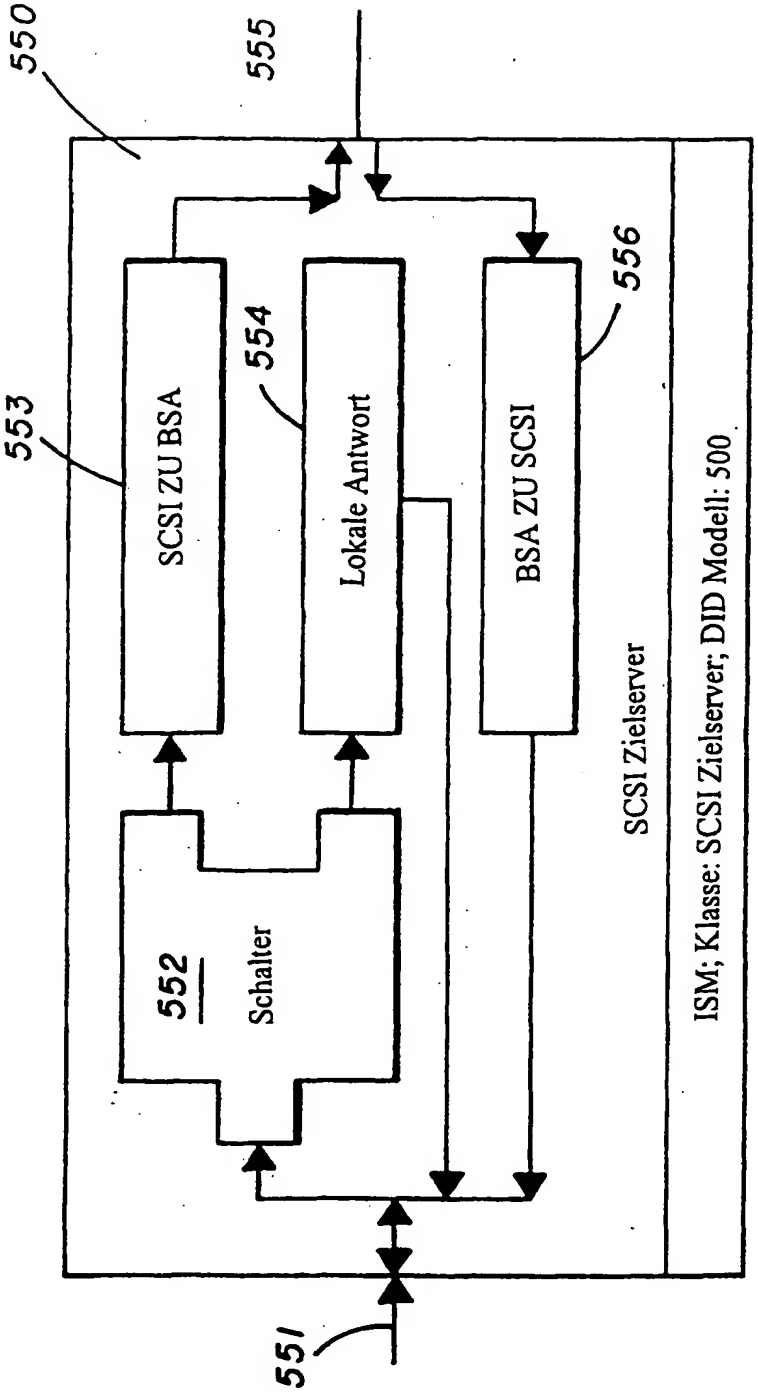


FIG. 11

Nummer: DE 100 14 448 A1
Int. Cl.[7]: G 06 F 15/173
Offenlegungstag: 16. November 2000

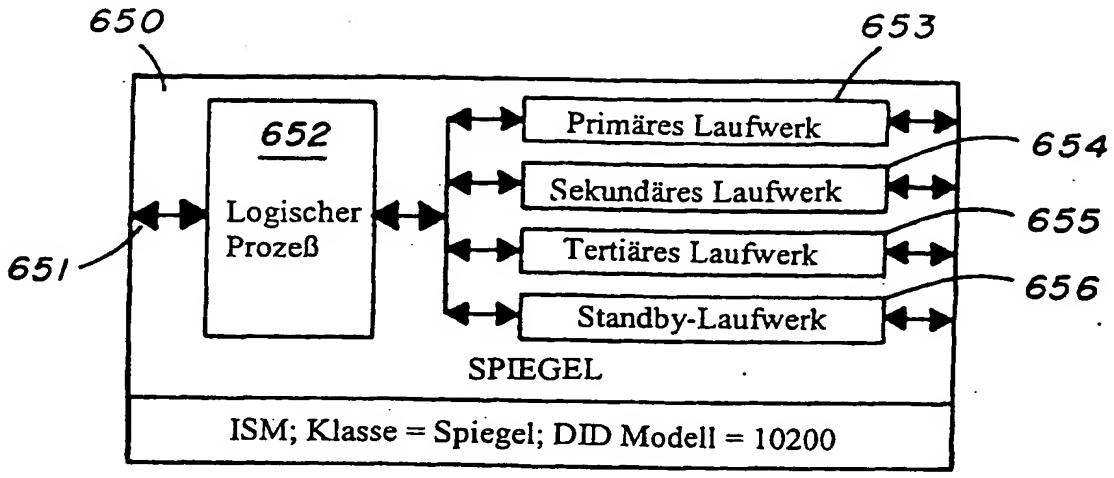


FIG. 12

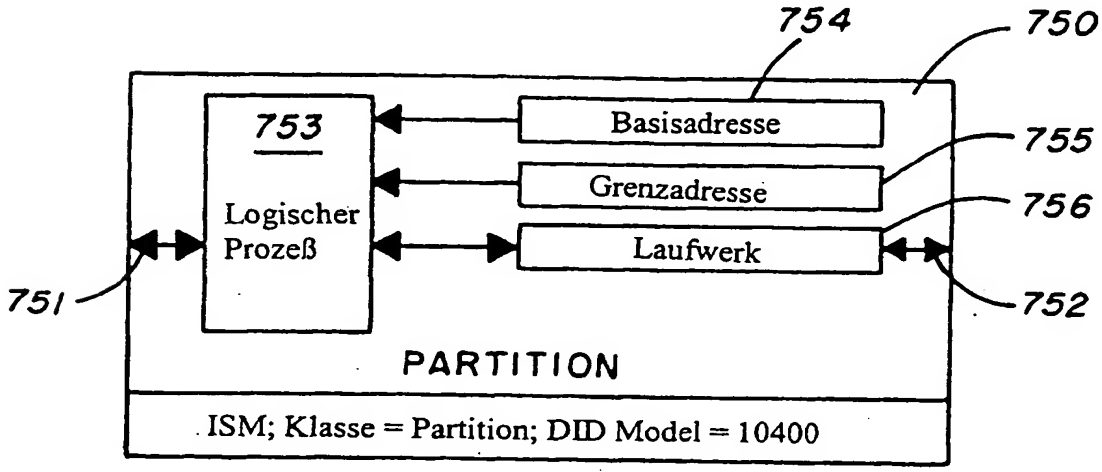


FIG. 13

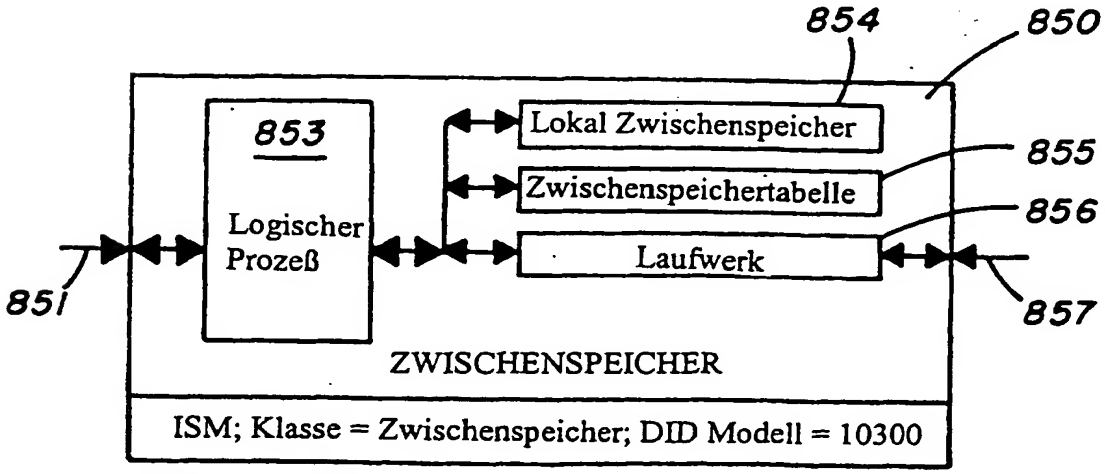


FIG. 14

Nummer: DE 100 14 448 A1
Int. Cl.[7]: G 06 F 15/173
Offenlegungstag: 16. November 2000

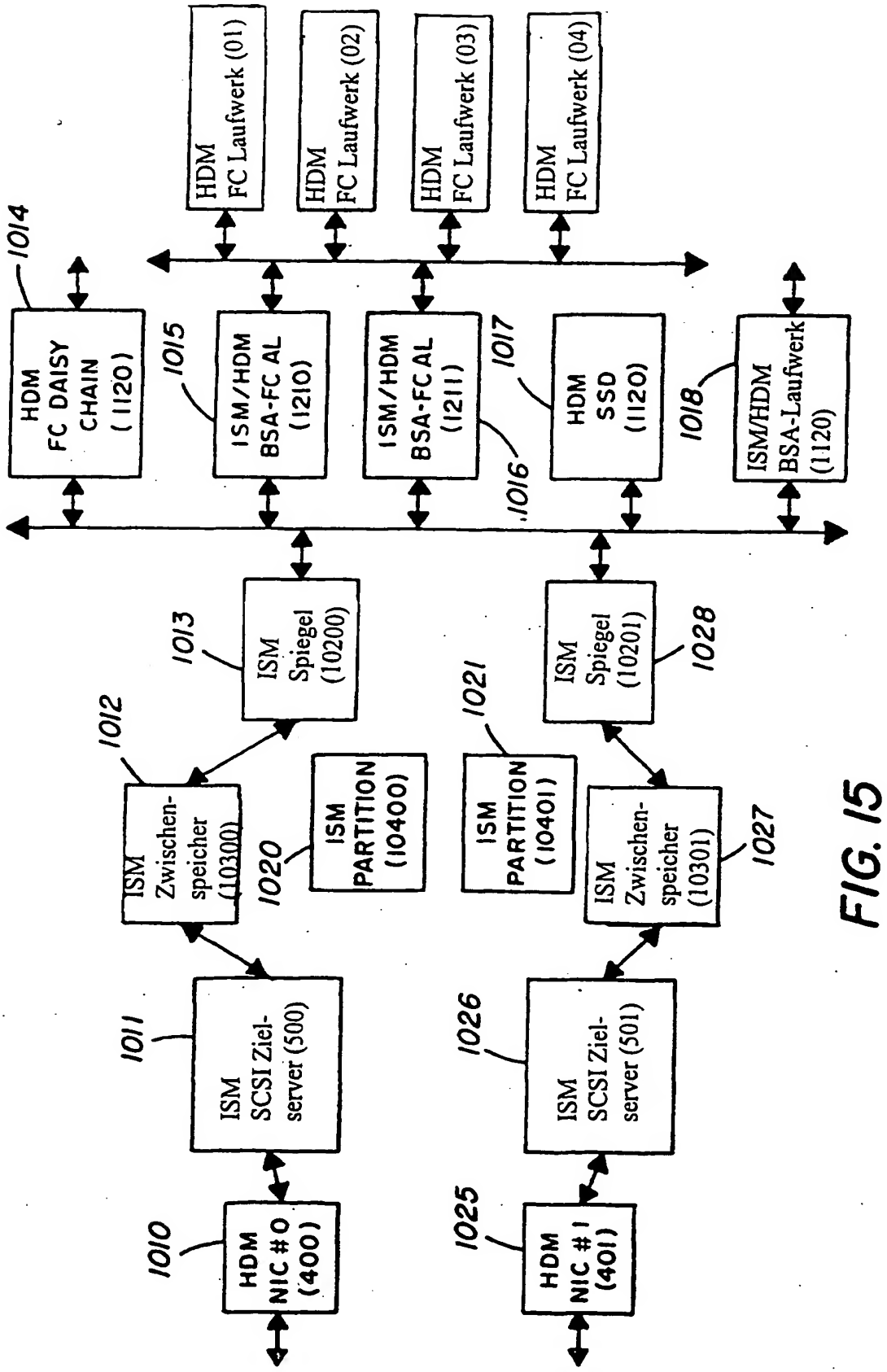


FIG. 15

Nummer: DE 100 14 448 A1
Int. Cl.[7]: G 06 F 15/173
Offenlegungstag: 16. November 2000

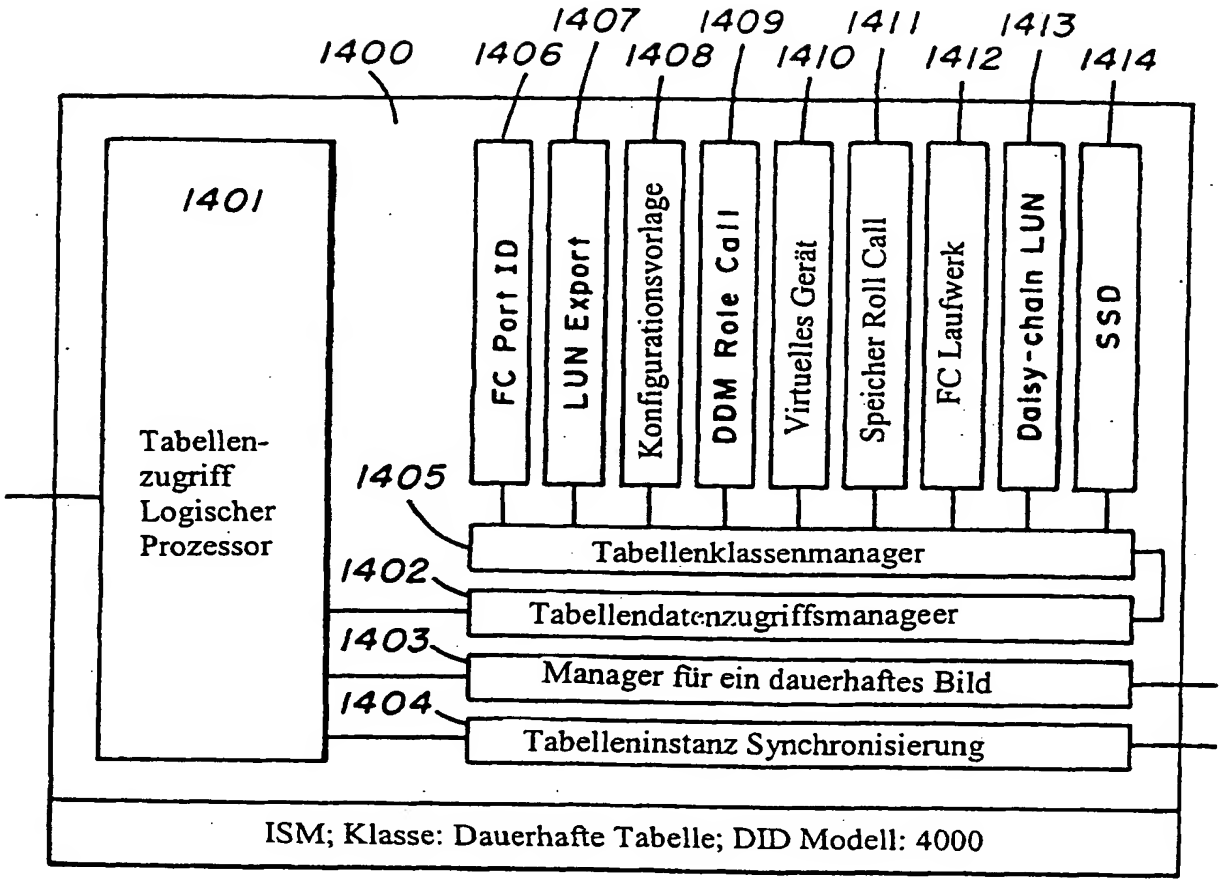


FIG. 16

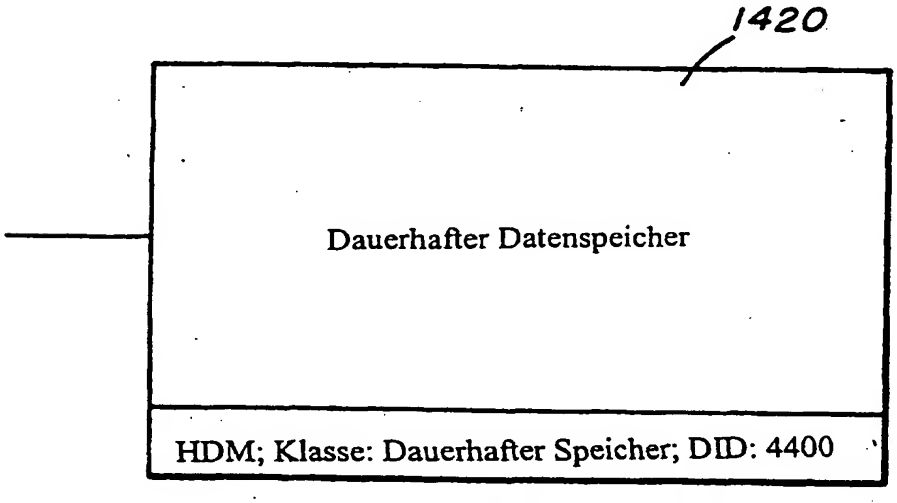


FIG. 17

Nummer: DE 100 14 448 A1
Int. Cl.[7]: G 06 F 15/173
Offenlegungstag: 16. November 2000

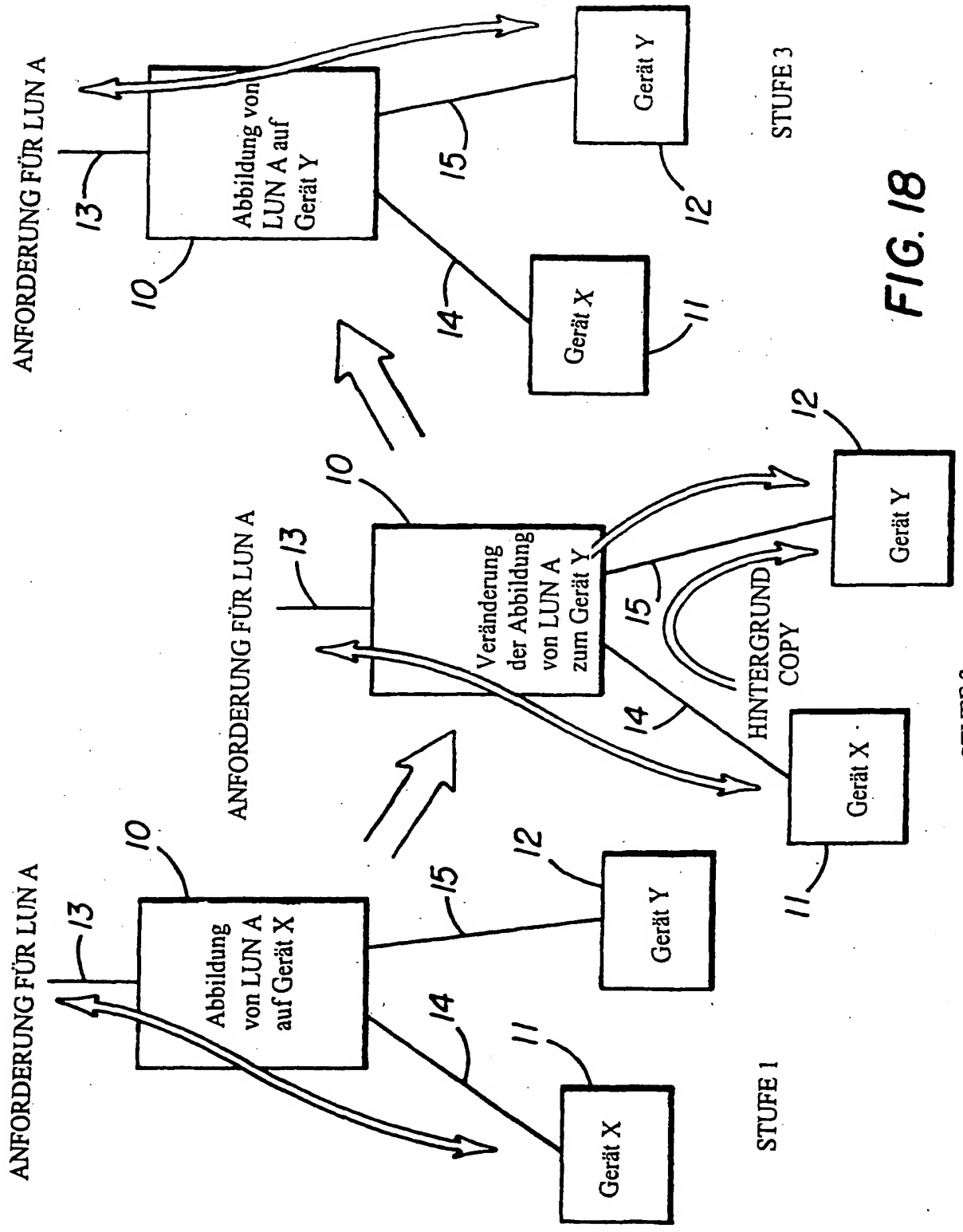


FIG. 18

Nummer: DE 100 14 448 A1
Int. Cl.[7]: G 06 F 15/173
Offenlegungstag: 16. November 2000

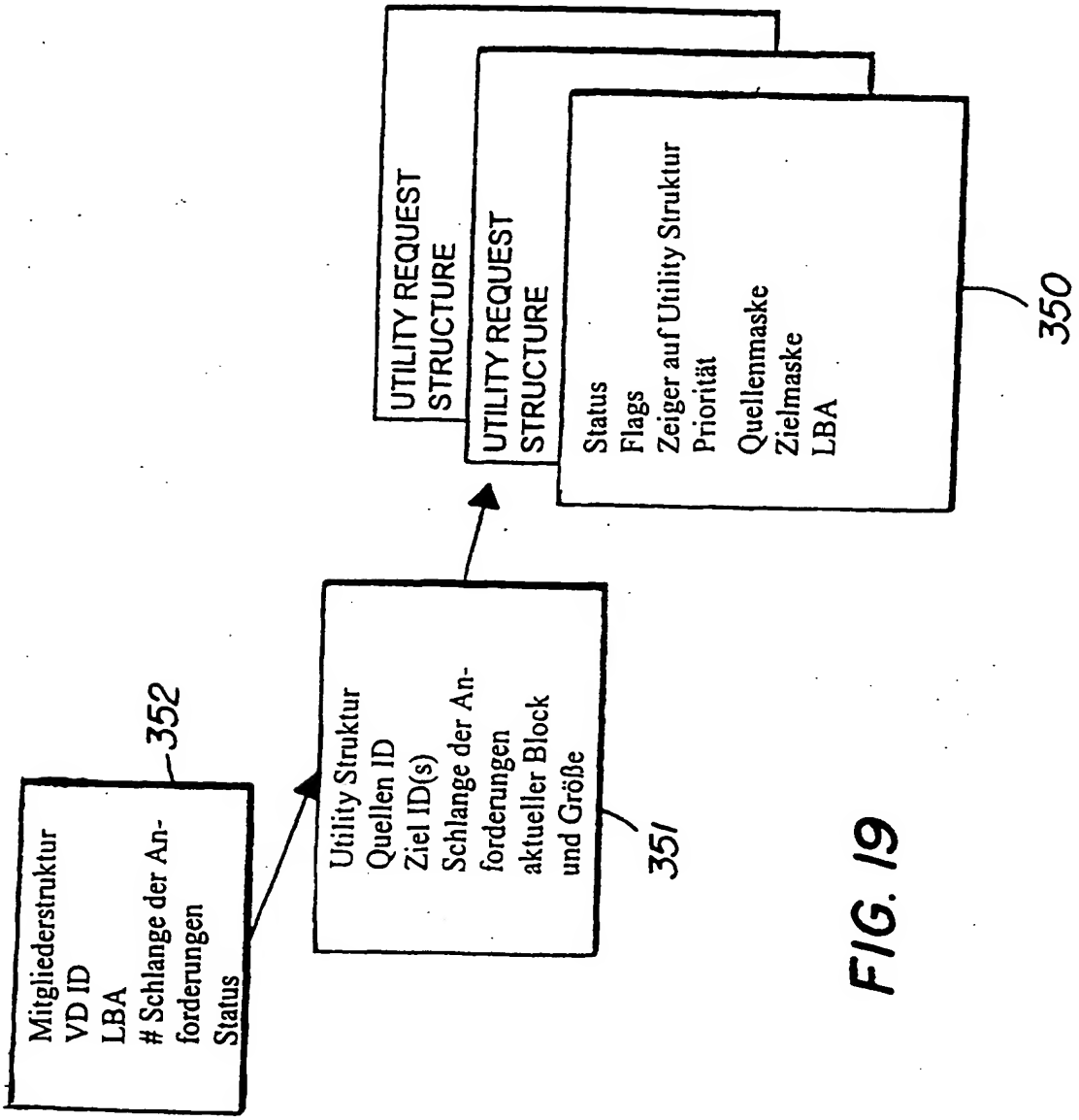


FIG. 19

Nummer: DE 100 14 448 A1
Int. Cl.[7]: G 06 F 15/173
Offenlegungstag: 16. November 2000

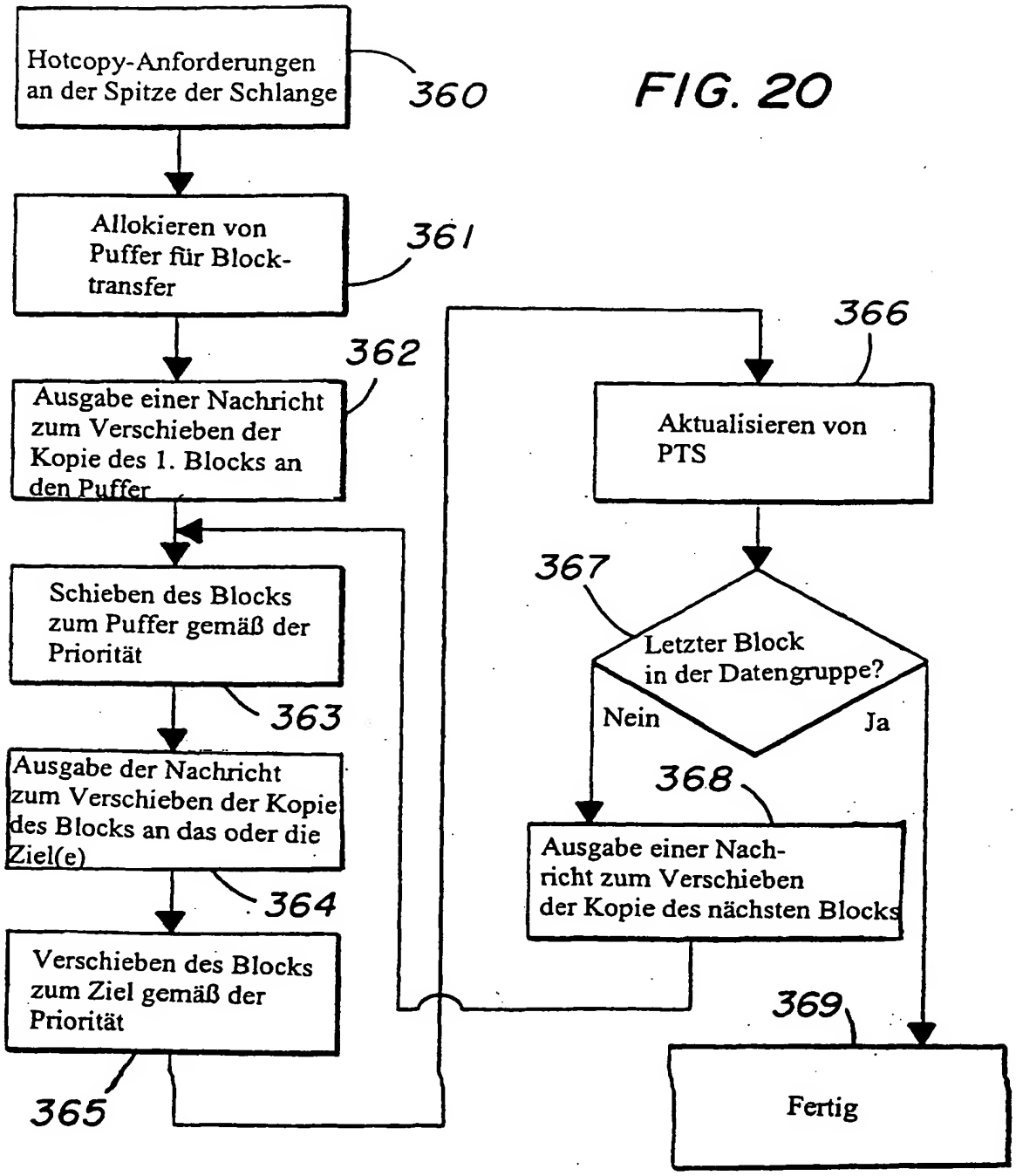


FIG. 20

Nummer: DE 100 14 448 A1
Int. Cl.[7]: G 06 F 15/173
Offenlegungstag: 16. November 2000

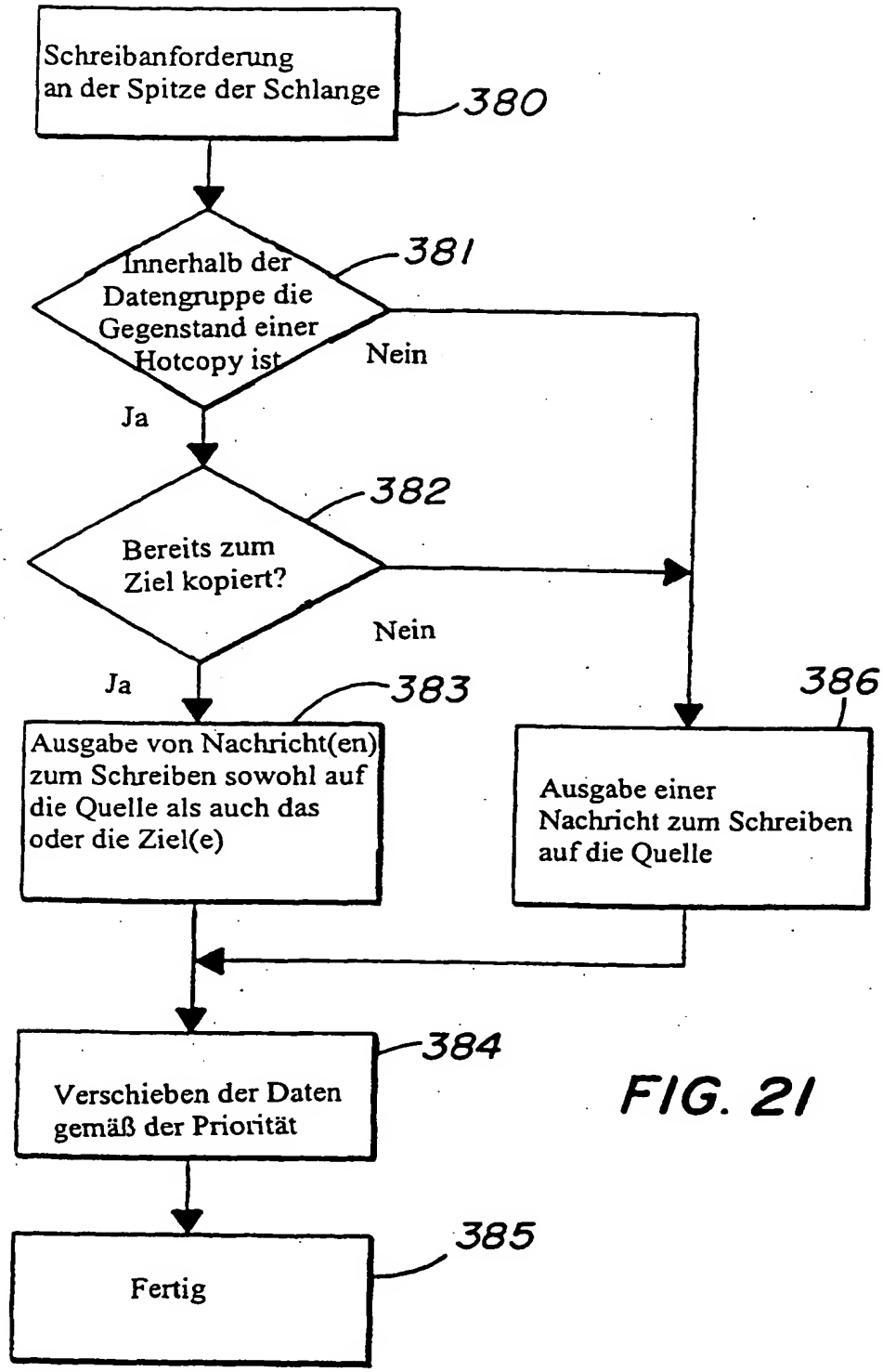


FIG. 21

Nummer: DE 100 14 448 A1
Int. Cl.[7]: G 06 F 15/173
Offenlegungstag: 16. November 2000

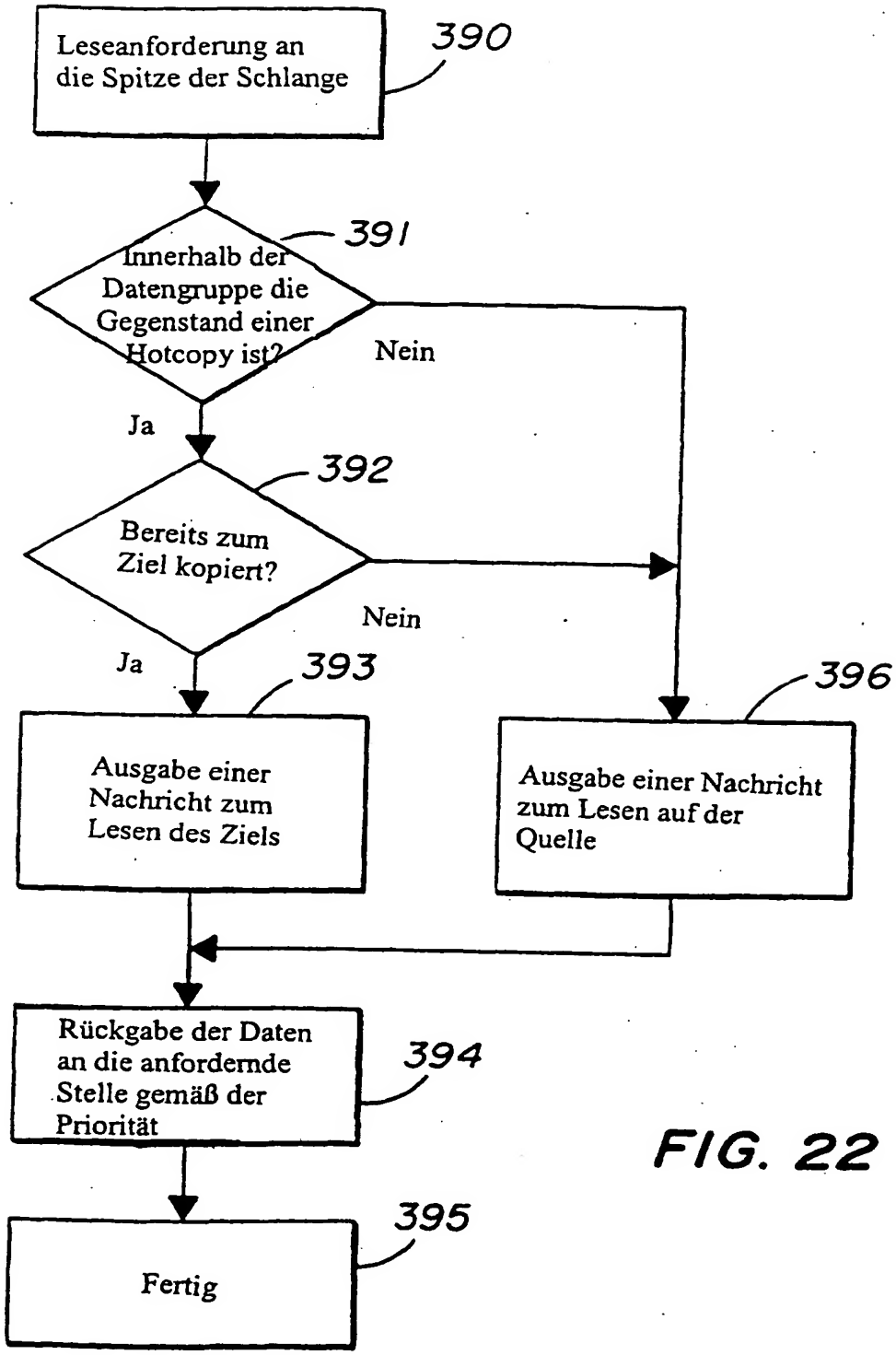


FIG. 22